



Universidad de Cuenca

Facultad de Ingeniería

Carrera de Electrónica y Telecomunicaciones

Análisis de amenazas de seguridad basado en la
detección de anomalías en el tráfico de red de la
infraestructura tecnológica de instituciones de
educación superior mediante el uso de técnicas de
machine learning

*Trabajo de titulación previo a la
obtención del título de Ingeniero en
Electrónica y Telecomunicaciones.*

Autor :

María José Vásquez Bravo
mjvb95@gmail.com

C.I. 010422949-7

Director :

Ing. Darwin Fabián Astudillo Salinas, PhD

C.I. 010390703-6

Cuenca - Ecuador
27 de Septiembre de 2021



Resumen

Las redes de comunicaciones han experimentado una evolución sin precedentes, debido principalmente a un aumento significativo del tráfico de datos. Esto convierte el tema de la seguridad de las infraestructuras tecnológicas en un punto importante a tratar dentro del ámbito de las [Instituciones de Educación Superior \(IES\)](#). Este tipo de instituciones manejan grandes cantidades de datos, que implican un aumento en el tráfico de red; por ello, el número de anomalías o vulnerabilidades han ido aumentando progresivamente. Estos ataques a la seguridad implican amenazas a la confidencialidad, integridad y/o disponibilidad de los datos manejados. Sin embargo, existen herramientas tales como los algoritmos de [Machine Learning \(ML\)](#), que permiten la detección previa de este tipo de eventos. En este marco, el presente trabajo experimental realiza la implementación de un *framework* que permita la detección de anomalías dentro del tráfico de red de las [IES](#) mediante la aplicación de técnicas de [ML](#), concretamente en el caso de la [Universidad Nacional del Chimborazo \(UNACH\)](#). Para ello, se analizó una recopilación de eventos correspondientes a un lapso de tiempo a través de la pila [Elasticsearch, Logstash y Kibana \(ELK\)](#); sometiéndolos a etapas de preprocesamiento, almacenamiento y visualización de datos para su análisis. A través de la aplicación del algoritmo K-Means, desarrollado mediante la librería Scikit-Learn de Python y el software Weka, se realizó un total de tres experimentos sobre los eventos recolectados. Esto permitió la detección de potenciales amenazas o anomalías, que serán presentadas y corroboradas mediante el uso de *dashboards* desarrollados en Kibana. A través de la implementación de este *framework*, se verificó la utilidad de los algoritmos de clusterización como herramienta óptima para la detección de anomalías dentro de una red universitaria. Obteniendo comportamientos anómalos dentro de la red tales como interferencias, solapamientos de canales, autenticación de usuarios no identificados o identificación de [Access Points \(AP\)](#) no autorizados.

Palabras clave : [IES](#). Anomalías. Amenazas. [ML](#). K-Means. Clústers. [ELK](#). Índices. Pipelines. *Dashboards*



Abstract

Communications networks have undergone unprecedented developments, mainly due to a significant increase in data traffic. This makes the issue of the security of technological infrastructures an important point to be addressed within the scope of Higher Education Institutions. These types of institutions handle large amounts of data, which imply an increase in network traffic; therefore, the anomalies or vulnerabilities number's have been progressively increasing. These security attacks involve threats to the confidentiality, integrity and/or availability of the data handled. However, there are tools such as [ML](#) algorithms, which allow pre-detection of such events. The present experimental work carries out the framework's implementation that allows anomalies' detection in the Higher Education Institutions network's traffic applying [ML](#) techniques, specifically in the [UNACH](#) case. To do this, a collection of events corresponding to a time lapse was analyzed through the [ELK](#) stack; subjected to data's preprocessing stages, storage and visualization for the analysis. Through the application of the K-Means algorithm, developed through Python's Scikit-Learn library and Weka software, a total of three experiments were performed on the collected events. This allowed the detection of potential threats or anomalies, which will be presented and corroborated using dashboards developed in Kibana. Through the implementation of this framework, the clusterization algorithms' utility was verified as an optimal tool for the anomalies' detection within a university network. Obtaining anomalous behaviors within the network such as interference, channel overlaps, authentication of unidentified users or identification of unauthorized [AP](#).

Keywords : Anomalies. Threats. [ML](#). K-Means. Clusters. [ELK](#). Index. Pipelines. Dashboards



Índice general

Resumen	1
Abstract	2
Índice general	3
Índice de figuras	6
Índice de tablas	8
Dedicatoria	11
Agradecimientos	12
Abreviaciones y acrónimos	13
1. Introducción	17
1.1. Identificación del problema	17
1.2. Justificación	17
1.3. Alcance	18
1.4. Objetivos	18
1.4.1. Objetivo general	18
1.4.2. Objetivos específicos	18
2. Marco teórico	19
2.1. Tráfico de red	19
2.2. Seguridad de redes y de la información	19
2.3. Machine Learning	21
2.3.1. Aprendizaje automático supervisado	21
2.3.2. Aprendizaje automático semi - supervisado	21
2.3.3. Aprendizaje automático no supervisado	21
2.4. Detección de anomalías	23
2.5. Pila ELK	23
2.5.1. Logstash	23
2.5.2. Elasticsearch	24
2.5.3. Kibana	25
2.6. WEKA	25



2.7. Scikit-Learn	25
3. Estado del arte	26
4. Diseño e implementación del framework	28
4.1. Diseño del <i>framework</i>	28
4.2. Implementación del sistema	29
4.2.1. Preprocesamiento del <i>Dataset</i>	30
4.2.2. Proceso de creación de índices e indexación de datos	31
4.2.3. Análisis de datos	31
4.2.4. Aplicación de técnicas de ML	31
4.2.5. Visualización de resultados	32
5. Evaluación del <i>framework</i>	33
5.1. Definición del escenario	33
5.2. <i>Dataset</i>	34
5.2.1. <i>Management Frames</i>	34
5.2.2. <i>Control y data Frames</i>	35
5.2.3. Rogue AP	35
5.2.4. IDS	36
5.2.5. RADIUS	37
5.2.6. <i>Warning</i>	37
5.3. Preprocesamiento del <i>dataset</i>	37
5.4. Proceso de creación de índices e indexación de datos	41
5.5. Dashboards	42
5.5.1. Análisis mediante Dashboards	42
5.6. Aplicación de técnicas de ML	45
5.6.1. Eventos de tipo <i>Rogue AP</i>	45
5.6.2. Eventos de tipo <i>Client Authenticated/Deauthenticated</i>	48
5.6.3. Relación entre <i>UserName</i> y <i>SSID</i>	50
5.7. Visualización de resultados	52
6. Conclusiones	56
6.1. Conclusiones	56
6.2. Trabajos Futuros	57
A. Instalación y configuración del <i>software</i>.	59
A.1. Instalación y configuración de la pila ELK	59
A.1.1. Prerrequisitos	59
A.1.2. Instalación y configuración de Elasticsearch	59
A.1.2.1. Edición del archivo de configuración principal de Elasticsearch	60
A.1.2.2. Iniciación del servicio Elasticsearch	60
A.1.3. Instalación y configuración de Logstash	60
A.1.3.1. Edición del archivo de configuración principal de Logstash	60
A.1.3.2. Iniciación del servicio Logstash	61



A.1.4. Instalación y configuración de Kibana	61
A.1.4.1. Edición del archivo de configuración principal de Kibana	61
A.1.4.2. Iniciación del servicio Kibana	61
A.1.5. Instalación y configuración de NGINX	62
B. Aprendizaje automático con Weka y creación de <i>dashboards</i>	64
B.1. Aprendizaje automático con Weka	64
B.2. Creación de <i>dashboards</i>	65
C. Dashboards de eventos identificados	66
Bibliografía	70



Índice de figuras

2.1. Identificación de una anomalía dentro de una señal [1].	20
2.2. Iteraciones del algoritmo de <i>K-Means</i> [2].	22
2.3. Infraestructura de la pila <i>ELK</i>	24
2.4. Estructura del archivo Pipeline.	24
4.1. Arquitectura general del <i>framework</i>	29
5.1. Topología de la red.	34
5.2. Planos de red.	35
5.3. Estructura de los eventos de tipo Client Authenticated/Deauthenticated.	35
5.4. Tipos de eventos <i>Rogue AP</i>	36
5.5. Estructura general de los <i>logs</i> capturados.	38
5.6. Fecha con formato “yyyy-MM-dd HH:mm:ss”.	38
5.7. Script de identificación y separación de variables.	39
5.8. Variables identificadas y delimitadas mediante “;”.	39
5.9. <i>Input</i> del pipeline de los eventos de tipo <i>Unauthorized Access Point (Rogue AP)</i>	40
5.10. <i>Filter</i> del pipeline de los eventos de tipo <i>Rogue AP</i>	40
5.11. <i>Output</i> del pipeline de los eventos de tipo <i>Rogue AP</i>	40
5.12. <i>Creación de los patrones de índice</i>	41
5.13. Listado de índices creados.	41
5.14. Visualización de datos en la línea temporal.	42
5.15. Estado de la variable username	43
5.16. Estado de los campos Estado y Channel.	44
5.17. Variables manejadas por los <i>logs</i> de tipo Warning	45
5.18. Número de clústers óptimo correspondiente a los eventos de tipo <i>Rogue AP</i>	46
5.19. Resultados obtenidos con cinco clústers para los logs de tipo <i>Rogue AP</i>	47
5.20. Número de clústers óptimo correspondiente a los eventos de tipo Client Authenticated/ Deauthenticated.	48
5.21. Resultados obtenidos con cuatro clústers para los eventos de tipo Client Authenticated/ Deauthenticated.	49
5.22. Número de clústers óptimo correspondiente a la relación entre las variables UserName y SSID.	50
5.23. Resultados obtenidos con tres clústers correspondientes a la relación entre las variables UserName y SSID.	51



5.24. Resultados obtenidos con cuatro clústers correspondientes a la relación entre las variables UserName y SSID.	52
5.25. <i>Dashboard</i> de resultados correspondientes al análisis de los eventos de tipo <i>Rogue AP</i>	54
5.26. <i>Dashboard</i> de resultados correspondientes al análisis de los eventos de tipo <i>Client</i> <i>Authenticated/Deauthenticated</i>	55
A.1. Ingreso a Kibana mediante el proxy inverso Nginx	63
B.1. Variables en la interfaz gráfica de Weka	64
B.2. K-Means en Weka	65
B.3. Interfaz gráfica de la herramienta Lens	65
C.1. Variables manejadas por los eventos de tipo Client Authenticated/Deauthenticated.	66
C.2. Variables manejadas por los <i>logs</i> de tipo IDSAttack.	67
C.3. Variables manejadas por los eventos de tipo RogueAP	68
C.4. Variables manejadas por los <i>logs</i> de tipo <i>Remote Access Dial in User Service</i> (RADIUS)	69



Índice de tablas

5.1. Tipos de eventos manejados en la administración de la conexión inalámbrica.	36
5.2. Códigos de deautenticación.	37



Cláusula de Propiedad Intelectual

María José Vásquez Bravo, autor/a del trabajo de titulación “Análisis de amenazas de seguridad basado en la detección de anomalías en el tráfico de red de la infraestructura tecnológica de instituciones de educación superior mediante el uso de técnicas de machine learning”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.

Cuenca, 27 de septiembre de 2021

María José Vásquez Bravo

C.I: 010422949-7



Cláusula de licencia y autorización para publicación en el Repositorio Institucional

María José Vásquez Bravo en calidad de autor/a y titular de los derechos morales y patrimoniales del trabajo de titulación “Análisis de amenazas de seguridad basado en la detección de anomalías en el tráfico de red de la infraestructura tecnológica de instituciones de educación superior mediante el uso de técnicas de machine learning”, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 27 de septiembre de 2021

María José Vásquez Bravo

C.I: 010422949-7



Dedicatoria

A mis padres **Víctor** y **Sheyla**, por todo su esfuerzo y apoyo tanto emocional como económico. A mi padre, por ser un ejemplo a seguir desde el día que tomó la decisión de cuidarme como su hija, guiarme, quererme, impulsarme y apoyarme a ser una profesional con ética y valores. A mi madre, por todo su cariño, amor, palabras de ánimo y enseñarme a ser una mujer fuerte que puede lograr sus metas a pesar de las dificultades.

A mis hermanos, **Daniela**, **Sebastián** y **Camilo**, por motivarme a ser un ejemplo para ellos y poner su confianza en mi y en mis habilidades. A mi hermana **Victoria**, que me cuida desde el cielo.

A mis abuelos, **Sixto** y **Gloria**, por todo su amor, palabras de ánimo y consejos.

A **Javier**, por todo lo vivido a lo largo de la carrera universitaria y fuera de ella, ser uno de mis mayores soportes en las etapas más difíciles y no dejarme decaer.

María José Vásquez B.



Agradecimientos

A mis padres, por su apoyo incondicional a lo largo de la etapa universitaria e impulsarme a cumplir mis metas y ser mi ejemplo para formarme como persona y profesional.

Al Ing. Darwin Fabián Astudillo Salinas, PhD, mi director de tesis, por su guía, apoyo y apertura a lo largo de la carrera universitaria y en el desarrollo del presente trabajo de titulación.

A mis compañeros, por las experiencias y enseñanzas inolvidables que vivimos día a día en las aulas de clase y fuera de ellas.

María José Vásquez B.



Abreviaciones y Acrónimos

AP *Access Points*. [1](#), [2](#), [6](#), [17](#), [23](#), [26](#), [34–37](#), [43](#), [45–48](#), [52](#), [56–58](#)

CEDIA Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia. [33](#), [49–51](#), [53](#)

CTS *Clear to send*. [35](#)

DDoS Denegación de Servicio Distribuido. [17](#)

DHCP Protocolo de configuración dinámica del servidor. [26](#)

DoS Denegación de Servicio. [17](#)

ELK Elasticsearch, Logstash y Kibana. [1](#), [2](#), [6](#), [18](#), [19](#), [23–25](#), [27–29](#), [56](#)

ESS *Extended Service Set*. [37](#)

GPG GNU Privacy Guard. [60](#), [61](#)

HTTP *Hypertext Transfer Protocol*. [62](#)

IBSS *Independent Basic Service Set*. [37](#)

IDS *Intrusion Detection System*. [27](#), [34](#), [36](#), [56](#), [57](#)

IES Instituciones de Educación Superior. [1](#), [19](#), [26](#), [28](#), [30](#), [32](#), [33](#), [49–52](#), [56](#)

IP *Internet Protocol*. [37](#), [61](#), [62](#)

MAC *Media Access Control*. [37](#), [47](#)

ML *Machine Learning*. [1](#), [2](#), [19](#), [21](#), [25–28](#), [30–32](#), [42–44](#), [57](#)

NIDS *Network Intrusion Detection System*. [27](#)

RADIUS *Remote Access Dial in User Service*. [7](#), [34](#), [37](#), [44](#), [69](#)

RAM *Random Access Memory*. [29](#), [30](#)

Rogue AP *Unauthorized Access Point*. [6](#), [7](#), [35](#), [40](#), [52](#), [54](#)

RTS *Request to Send*. [35](#)

SVM *Support vector machine*. [25](#)

TCP *Transmission Control Protocol*. [26](#)

UIT Unión Internacional de Telecomunicaciones. [17](#)

UNACH Universidad Nacional del Chimborazo. [1](#), [2](#), [33](#), [37](#), [42](#), [43](#), [47](#), [49](#), [57](#)

UTC *Coordinated Universal Time*. [38](#)



WEKA *Waikato Environment for Knowledge Analysis.* [25](#)
WEP *Wired Equivalent Privacy.* [26](#)
WLAN *Wireless Local Area Networks.* [19](#), [26](#), [28](#), [34](#), [36](#)
WLC *Wireless LAN Controller.* [28](#), [33](#), [34](#), [56](#)
WPA *Wi-Fi Protected Access.* [26](#)



Introducción

Este capítulo presenta la identificación del problema, justificación y los objetivos tanto general como específicos del presente proyecto.

1.1. Identificación del problema

En las instituciones de educación superior el número de usuarios de la red ha ido aumentando de manera progresiva. Estos usuarios disponen de servicios de red inalámbrica, cableada, entre otros, siendo la principal fuente de tráfico en la red de la institución [3]. Como consecuencia, las amenazas y vulnerabilidades a las que se enfrentan los sistemas han aumentado considerablemente, siendo los problemas más comunes la detección de *spam*, páginas de *phishing*, ataques [Denegación de Servicio \(DoS\)](#) y [Denegación de Servicio Distribuido \(DDoS\)](#), detección de anomalías, entre otros [4], poniendo en riesgo la confidencialidad, disponibilidad e integridad de los datos que manejan [5]. Por ello, estas deben ser consideradas a la hora de crear *frameworks* para que permitan la protección de los datos de posibles amenazas en el ciberespacio.

1.2. Justificación

La [Unión Internacional de Telecomunicaciones \(UIT\)](#) en su Recomendación UIT-T X.1205 [6], define ciberseguridad como “*el conjunto de herramientas, políticas, conceptos de seguridad, salvaguardas de seguridad, directrices, métodos de gestión de riesgos, acciones, formación, prácticas idóneas, seguros y tecnologías que pueden utilizarse para proteger los activos de la organización y los usuarios en el ciberentorno. Los activos de la organización y los usuarios son los dispositivos informáticos conectados, los usuarios, los servicios/aplicaciones, los sistemas de comunicaciones, las comunicaciones multimedia, y la totalidad de la información transmitida y/o almacenada en el ciberentorno.*”. Los diferentes equipos de red, tales como [AP](#), *switch*, *firewalls*, controladores, entre otros, generan diariamente una gran cantidad de *logs* correspondientes a la actividad producida en la comunidad universitaria. Estos *logs* son archivos de texto simples que registran las actividades de los usuarios cada vez que requieren un

recurso de la red [7]. Si son gestionados y procesados correctamente, los *logs* entregan información útil para el análisis y posterior toma de decisiones.

1.3. Alcance

El presente trabajo de titulación plantea la aplicación de técnicas de *machine learning* con la finalidad de obtener conocimiento sobre el comportamiento de la red y los usuarios en las instituciones de educación superior. El conocimiento obtenido ayudará a predecir y detectar anomalías en el tráfico de la red. El proyecto se desarrollará en el marco del Grupo de Trabajo de Ciberseguridad de CEDIA y con el apoyo de la Universidad de Cuenca y la Universidad Nacional del Chimborazo. El proceso de detección de anomalías se encuentra dividido en diferentes etapas: recolección de datos, preprocesamiento, indexación de datos, análisis de resultados, aplicación de la técnica de *machine learning* seleccionada, visualización de resultados [8]. Para llevar a cabo el proceso relacionado con la recolección de datos, preprocesamiento, indexación y análisis de los *logs* se plantea el uso de las herramientas *open source* más relevantes en la actualidad, en este caso la pila [ELK](#); esta pila está formada por las herramientas Elasticsearch, Logstash y Kibana. Una vez finalizadas estas etapas, se procederá a la aplicación de la técnica de *machine learning* seleccionada mediante la herramienta *open source* Weka y la librería SciKit Learn de Python. Weka y SciKit Learn permitirán la implementación y aplicación del algoritmo en función del tipo de experimento; el *framework* implementado entregará como resultado las posibles anomalías detectadas en el tráfico de red; y además, estos resultados serán presentados en un *dashboard* desarrollado en Kibana.

1.4. Objetivos

1.4.1. Objetivo general

Utilizar técnicas de *machine learning* para la detección de anomalías y posterior análisis de amenazas de seguridad en el tráfico de la red de instituciones de educación superior.

1.4.2. Objetivos específicos

El presente trabajo tiene los siguientes objetivos específicos:

- Realizar un estado del arte sobre la detección de anomalías y las herramientas usadas en este ámbito.
- Obtener los *logs* generados de diversos equipos de la red de la Universidad.
- Implementar el *framework* para la detección de anomalías y análisis de amenazas de seguridad en el tráfico del entorno universitario.
- Aplicar técnicas de *machine learning* tales como asociación, clasificación, regresión o agrupación, según sea el caso de al menos 3 experimentos.
- Implementar un *dashboard* que permita presentar los resultados obtenidos.



CAPÍTULO 2

Marco teórico

En este capítulo se presenta una serie de conceptos que permitirán un mejor entendimiento del trabajo propuesto, tales como: tráfico, seguridad en las redes inalámbricas, y el tipo de posibles ataques y anomalías que pueden sufrir las redes de las [IES](#). Además, se presenta un breve marco teórico relacionado con la detección de anomalías y el [ML](#), realizando una introducción a este campo y a los algoritmos a utilizar. Finalmente, se describe las herramientas utilizadas a lo largo del procedimiento, tales como: [ELK](#), Scikit Learn y Weka.

2.1. Tráfico de red

El tráfico de red se define como la cantidad de datos que son enviados por una red en un momento específico. Para enviar los datos, estos se dividen en paquetes, los cuales son dirigidos a través de la red y ensamblados en el dispositivo receptor. El tráfico de datos tiene una relación directa con la calidad de la red, ya que puede influenciar en las velocidades de subida o bajada de datos, conexiones irregulares o incluso un tráfico de red elevado que implique cambios en su rendimiento; como el número de conexiones por segundo (c/s), conexiones simultáneas máximas (mcc) o transacciones por segundo (t/s), situaciones que pueden indicar posibles ataques [\[9\]](#).

Las redes inalámbricas generan tráfico de red, siendo una red inalámbrica un tipo de conexión entre dispositivos o terminales móviles, donde la información es transmitida mediante ondas electromagnéticas. En la actualidad, las [Wireless Local Area Networks \(WLAN\)](#) han obtenido gran relevancia debido a sus prestaciones y gran velocidad de transmisión de datos, además de su robustez y desarrollo continuo [\[10\]](#).

2.2. Seguridad de redes y de la información

El concepto de seguridad en redes y la información, consiste en la protección de posibles accesos, uso, alteración o destrucción no autorizada [\[1\]](#). En la actualidad, las [IES](#) manejan su información mediante redes informáticas. Estas redes no se ven exentas de posibles fallas internas debido a su

complejidad o la dificultad de detectar y corregir problemas de seguridad. Además, pueden presentar ataques externos por parte de terceros, lo cual debe ser considerado a la hora de crear un *framework* para la protección de sus datos [11]. En este contexto, existen conceptos claves ligados a los ataques informáticos, citados a continuación:

Vulnerabilidad: Las vulnerabilidades dentro de un sistema surgen a partir de errores individuales o colectivos. Estos errores generan problemas de seguridad para la red. Una de las vulnerabilidades más conocidas es la escalada de privilegios (ejecución de programas con privilegios de administrador) o generación de error de sistema [12].

Amenaza: Se la define como acción que forma parte de una posible causa de riesgo para una persona o entorno. Las principales amenazas a la seguridad son: *Malware*, *Spamming* y las utilizadas para comprometer la seguridad de los sistemas [12].

Riesgo: Se define como la “posibilidad de que se produzca un contratiempo o que algo o alguien sufra perjuicio o daño” [12]. Los recursos que manejan las redes pueden ser expuestas a diversos riesgos, los cuales están relacionados con confidencialidad, integridad y disponibilidad.

Anomalía: Se define anomalía como todo aquello que se escapa de lo esperado, es decir, no se definen por sus propias características sino por la contraposición de lo que sí es normal [13]. En la Figura 2.1, se presenta el rastreo de una anomalía dentro de la captura de una señal, visualizando el comportamiento normal de la señal (2.1b) frente a su estado con la anomalía detectada (2.1c). La detección de anomalías en la red consiste en identificar comportamientos o valores atípicos que se alejan del comportamiento normal del tráfico de red, por lo que tiene relación directa con el modelado del tráfico [1].

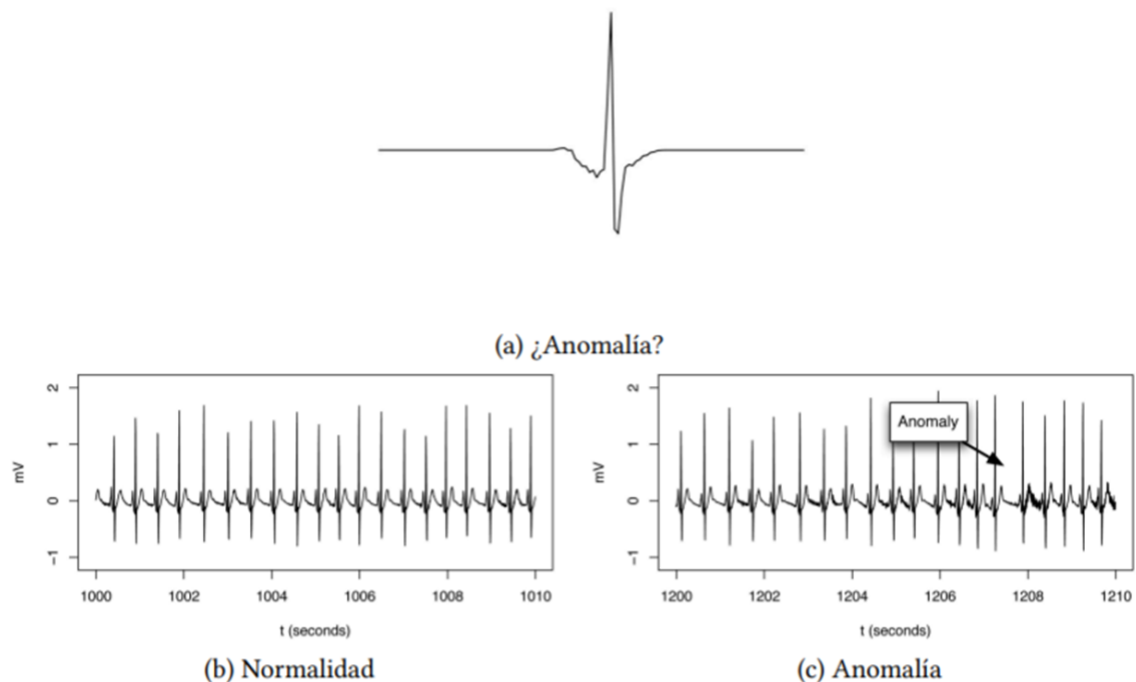


Figura 2.1: Identificación de una anomalía dentro de una señal [1].

2.3. Machine Learning

Se denomina aprendizaje como la “Adquisición de conocimiento de algo por medio del estudio, el ejercicio o la experiencia” [14], lo que permite desarrollar o resolver de mejor manera una tarea a futuro. Dentro del ámbito de la inteligencia artificial, el aprendizaje automático o ML se centra en el desarrollo de algoritmos que puedan acceder y manipular los datos con la finalidad de aprender por sí mismos. Para ello se lleva a cabo un análisis de la información con la finalidad de reconocer patrones y tomar decisiones a futuro, evitando así la intervención humana [15].

2.3.1. Aprendizaje automático supervisado

Este tipo de aprendizaje se basa en el manejo de un conjunto de datos, siendo una técnica para derivar funciones a partir de datos de entrenamiento. Los datos de entrenamiento constan de pares de objetos: un componente del par son los datos de entrada (x) y el otro es el resultado deseado (y), (Ecuación 2.1) que permiten la obtención de algoritmos que clasifiquen datos o predigan resultados mediante el mapeo de la entrada a la salida [16].

$$y = f(x) \tag{2.1}$$

El objetivo principal es la aproximación de la función de mapeo para su utilización con nuevos datos y la entrega de resultados más precisos. Este tipo de aprendizaje se maneja mediante un conjunto de datos de entrenamiento que supervisa el proceso de aprendizaje realizando pruebas de manera iterativa y ajustando ponderaciones hasta que el modelo se detiene, consiguiendo así un valor de error mínimo y un nivel aceptable de rendimiento. Los problemas de aprendizaje supervisado se pueden dividir en dos grupos: clasificación y regresión [16].

2.3.2. Aprendizaje automático semi - supervisado

Este tipo de aprendizaje combina datos etiquetados con datos no etiquetados con la finalidad de generar el modelo deseado. Los algoritmos aprenden las estructuras para organizar datos y realizar predicciones.

2.3.3. Aprendizaje automático no supervisado

Este tipo de aprendizaje posee datos de entrada (x) y no posee las variables de salida, por lo que el conjunto de datos no se encuentra etiquetado y los resultados no son conocidos. A diferencia del aprendizaje supervisado, los algoritmos pertenecientes a este grupo generan su propio diseño y devuelve la información relevante de los datos. Para este tipo de algoritmos es necesaria la deducción de las estructuras presentes en el *dataset* de entrada; esto se consigue mediante la aplicación de procesos matemáticos con la finalidad de reducir la redundancia u organizar los datos por similitud. Los problemas de aprendizaje no supervisado se pueden dividir en: agrupación (clusterización) y asociación.

En este proyecto se va a utilizar únicamente la clusterización, en concreto el algoritmo de K-Means, como se presenta a continuación. La clusterización o algoritmos de agrupamiento, es uno de los métodos de identificación de patrones usados dentro del aprendizaje automático o ML. Los algoritmos pertenecientes a esta categoría, basan su funcionamiento en la identificación de similitudes

entre vectores, agrupándolos en clústers. Este método se enmarca dentro de la minería de datos como aprendizaje no supervisado.

- K – Means: Tiene como objetivo descubrir agrupamientos dentro de un *dataset* basándose en sus características. Este algoritmo es un método no jerárquico basado en centroides y clústers que consta de etapas diferenciadas:

1. Seleccionar la partición inicial del *dataset* en k clúster C_1, \dots, C_k .
2. Calcular los centroides de cada clúster mediante la Ecuación 2.2.

$$w_l = \frac{1}{k_i} \sum_{j=1}^{k_i} w_{ij}, i = 1, \dots, k \quad (2.2)$$

3. Por cada w_i en la base de datos y siguiendo el orden de las instancias:
 - a) Reasignación de la instancia w_i al centroide del clúster más cercano.
 $w_i \in C_s$ es movido desde C_s hasta C_t si $\|w_i - \bar{w}_t\| \leq \|w_i - \bar{w}_j\|$ para todo $j = 1, \dots, k$
 $j \neq s$
 - b) Recalcular los centroides para los clústeres C_s y C_t .
4. Reasignar nuevamente los elementos del *dataset* al centroide más cercano.
5. Si el clúster se encuentra estabilizado, el proceso habrá finalizado, caso contrario, se regresará al paso 3 [2]. En la Figura 2.2 se visualiza las iteraciones realizadas por el algoritmo, hasta obtener finalmente una estabilización completa de los clústers.

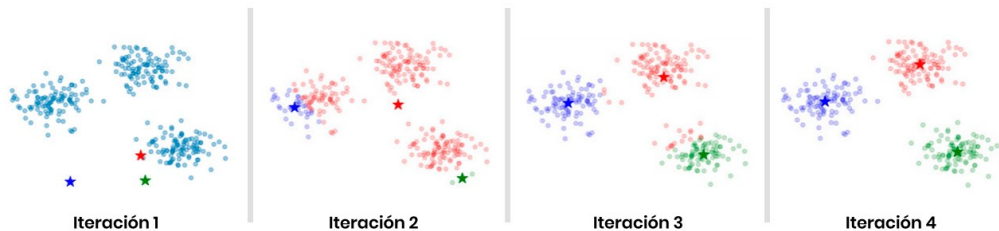


Figura 2.2: Iteraciones del algoritmo de *K-Means* [2].

■ Selección del número de clústers: Método de Elbow

Para implementar el algoritmo de *K-Means*, es necesario seleccionar de manera correcta el número de clústers. Para ello, existe el método de Elbow o método del “codo”, el cual se basa en la suma de los cuadrados de las distancias de cada elemento con el centroide que le corresponde (Ver Ecuación 2.3).

Donde:

$$WCSS = \sum_{i \in n} (X_i - Y_i)^2 \quad (2.3)$$

- ◇ $WCSS$: Suma de los cuadrados de las distancias.
- ◇ X_i : Elemento o dato
- ◇ Y_i : Centroide del elemento o dato.
- ◇ n : Total de datos de la muestra

El proceso se inicia con un solo clúster para todos los elementos, calculando la suma de las distancias de cada elemento con el centroide. A continuación, se repite el proceso para n centroides, sumando las distancias de los elementos más cercanos a sus centroides correspondientes, y finalizando el proceso al obtener una suma de distancias igual a cero, dado que cada elemento es un centroide [17].

2.4. Detección de anomalías

La detección de anomalías en el tráfico de red es fundamental para la eliminación oportuna de eventos como posibles ataques o fallas, que pueden afectar a factores relacionados con la seguridad o rendimiento de la red, como:

- Velocidad de internet o ancho de banda: Mide la cantidad de información que es capaz de transportar por unidad de tiempo. Su unidad de medida es *bits* por segundo (bps).
- Latencia: Tiempo que transcurre desde que un *byte* es enviado hasta que llega al otro extremo.
- Conexiones por segundo: Velocidad en la que un dispositivo puede establecer parámetros para nuevas conexiones.
- Transacciones por segundo: Número de acciones completas que se pueden realizar en un segundo.
- Conexiones simultáneas máximas: Número total de sesiones que un dispositivo puede mantener simultáneamente.

Numerosos estudios se han centrado en esta problemática, por ejemplo, en [18] los autores se centran en la evaluación de las señales anómalas en el tráfico en diferentes puntos dentro de una red, basándose en la distancia topológica desde el origen al destino de las anomalías. Por otro lado, en [19] se centra en la detección de anomalías basadas en características de tráfico, llevando a cabo el modelado de patrones mediante histogramas de diferentes características. Ambos documentos indican la eficacia de la detección de anomalías en el tráfico de red, entregando modelos y resultados que sirven como pilar para futuros trabajos.

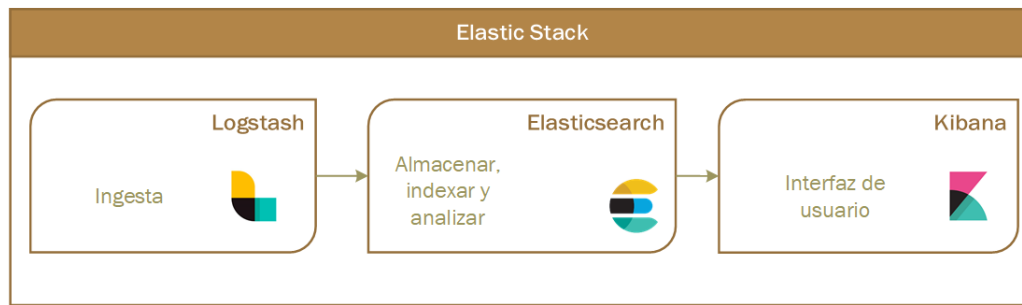
En la actualidad, se dispone de un mayor número de métodos y herramientas que permiten la detección de comportamientos anómalos en la red, como las técnicas de aprendizaje automático donde su enfoque posee altas capacidades para aprender y reconocer patrones complejos, con la posibilidad de tomar decisiones inteligentes basadas en datos [20].

2.5. Pila ELK

La pila [ELK](#) es un grupo de herramientas *open source* que trabajan de manera conjunta para procesar, monitorizar y analizar *logs* generados por diferentes equipos de red [21], tales como [AP](#), *switch*, *firewalls*, controladores, entre otros. Estos equipos generan diariamente una gran cantidad de *logs* correspondientes a la actividad producida en diversos entornos. Los componentes que conforman este grupo de herramientas son: Logstash, Elasticsearch y Kibana (ver Figura 2.3).

2.5.1. Logstash

Para la definición de la herramienta Logstash, es necesario conocer definiciones previas tales como *pipeline*. Se define *pipeline* como la canalización utilizada para métodos de codificación y automatización

Figura 2.3: Infraestructura de la pila [ELK](#).

del flujo de trabajo necesarios para generar un modelo. Los *pipelines* de ingesta de datos permiten realizar transformaciones en estos, como la eliminación de campos o extracción de valores de texto, entre otros, antes de proceder con el almacenamiento o indexación. En la Figura 2.4 se presenta la estructura manejada por estos archivos, donde se visualiza una serie de tareas configurables denominadas procesadores que se ejecutan de forma secuencial y realiza cambios específicos en los documentos entrantes. Estos archivos de configuración deben contener al menos un *plugin* de entrada (*input*), un *plugin* de salida (*output*) y una etapa de filtrado o *filter* encargada de realizar los procesos de transformación, normalización o enriquecimiento.

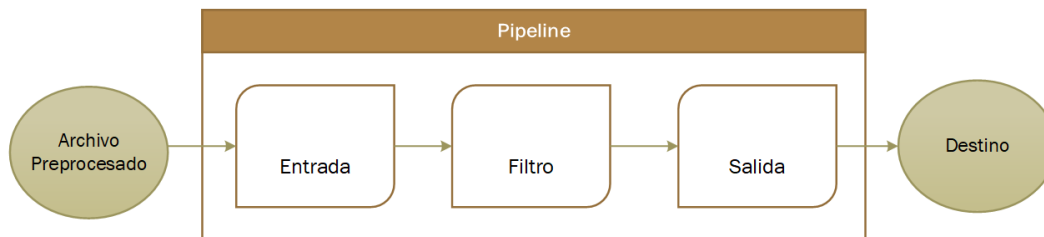


Figura 2.4: Estructura del archivo Pipeline.

Dentro de este contexto, se define Logstash como un *pipeline* de procesamiento de datos del lado del servidor que permite la ingesta de datos de manera centralizada, independientemente de su formato o complejidad. Esta herramienta toma la información generada por diferentes equipos de red, los transforma y enruta a la ubicación deseada, en este caso Elasticsearch [22].

2.5.2. Elasticsearch

Elasticsearch es el núcleo de la pila [ELK](#) y se encarga de almacenar datos como documentos JSON. Se trata de una base de datos distribuida tolerante a fallos, encargada de almacenar la información en diversos nodos y de distribuir el procesamiento [23]. Esta herramienta trabaja mediante índices, que son definidos como una “base de datos” dentro de una base de datos relacional, es decir, permite al usuario la organización de datos mediante la asignación de nombres que permiten su identificación y relación con las réplicas y fragmentos distribuidos por el clúster. Estas agrupaciones de documentos mantienen relación entre sí, manejando conjuntos de claves con sus valores correspondientes [24].



2.5.3. Kibana

Kibana es una herramienta que actúa como interfaz de usuario, permitiendo la gestión y visualización de datos que se encuentran indexados en Elasticsearch. La Visualización se realiza mediante la creación de diversos tipos de gráficas (barras, circulares, histogramas, etc.) y dashboards. Además, monitorea y administra una instancia de [ELK](#) usando la interfaz web, centralizando el acceso para soluciones integradas [\[25\]](#).

2.6. WEKA

El entorno de Waikato para el análisis de conocimiento, denominado *Waikato Environment for Knowledge Analysis (WEKA)*, es un *software* de recopilación de las técnicas más avanzadas de [ML](#) [\[26\]](#) y minería de datos desarrollada en Java. Algunas de las opciones que ofrece esta herramienta son [\[27\]](#):

- Preprocesamiento de datos y visualización.
- Selección de atributos.
- Reglas de asociación.
- Algoritmos de clasificación.
- Algoritmos de Predicción.
- Técnicas de Evaluación
- Clusterización.

2.7. Scikit-Learn

Scikit-Learn es una librería de Python, relacionada con el campo de [ML](#). Esta librería permite la implementación de algoritmos de aprendizaje supervisados, tales como regresión lineal, *Support vector machine (SVM)*, árboles de decisión, entre otros. Además permite la implementación de algoritmos de aprendizaje no supervisado, varios conjuntos de datos o *dataset* para la realización de pruebas o la extracción y selección de características de imágenes y texto [\[28\]](#).

Esta librería está edificada sobre *Scientific Python* e incluye las librerías adicionales citadas a continuación [\[28\]](#):

- *NumPy*: Librería de manejo de matrices de n dimensiones.
- *Pandas*: Manejo de estructura de datos y análisis.
- *Matplotlib*: Librería encargada de trazados en dos dimensiones.



Estado del arte

Toda institución que ofrece un servicio tiene como responsabilidad asegurar que la información manejada no será extraída o manipulada por parte de terceros no autorizados, siendo la seguridad en el entorno institucional como empresarial un punto de vital importancia a la hora de crear un *framework* que permita salvaguardar los datos o información. Dentro de este contexto, las IES manejan día a día una gran cantidad de datos relacionados con el tráfico de red generado por autoridades, personal administrativo, cuerpo docente o estudiantes, por lo que el número de anomalías, amenazas a la seguridad o vulnerabilidades han ido aumentando paralelamente. Para contrarrestar los posibles ataques antes mencionados, es necesario conocer las anomalías más frecuentes en la red, los tipos de datos manejados y los métodos y sistemas de detección actuales [29]. El campo de la ciberseguridad es el encargado de la seguridad en redes, aplicaciones e información [6]; que junto con herramientas externas, tales como los algoritmos de ML, permiten detectar posibles estados anómalos en la red. En el presente apartado se realiza una revisión exhaustiva de investigaciones que tratan el tema relacionado a la detección de anomalías dentro de la red, entregando conclusiones y resultados que sirven como soporte para el presente trabajo de titulación o trabajos futuros.

En [30], Vallejo presentó un análisis sobre los ataques más comunes dentro de las WLAN tales como ataques pasivos (Warchalking, Wardriving, Sniffing), activos (Enmascaramiento o suplantación, secuestro de sesión, suplantación de dirección MAC) y problemas concretos junto a una serie de métodos para reducir la inseguridad. Como resultado se obtuvo una desventaja considerable en la redes inalámbricas, ya que las ondas de radiofrecuencia pueden ser interceptadas y analizada por parte de terceros. Además, existe la utilización de programas diseñados para la manipulación de datos, tales como: programas falsificadores de AP, analizadores de tráfico de red, programas descriptores falsificadores de direcciones MAC. Estos ataques pueden ser contrarrestados mediante autenticación de usuario, claves de acceso a la red, listas de control de acceso en puntos de acceso y cifrado *Wired Equivalent Privacy* (WEP) o *Wi-Fi Protected Access* (WPA). Dentro de este ámbito, en [31] se realizó el estudio puntual de los ataques de red de tipo: *Protocolo de configuración dinámica del servidor* (DHCP) *spoofing*, *Transmission Control Protocol* (TCP) *SYN flood* y paquetes malformados, los cuales pueden capturar mensajes o afectar al rendimiento del ancho de banda. Este proceso se realizó mediante

el análisis del tráfico circulante en una intranet, a través de la emulación de subredes. Esto indicó la factibilidad del uso de herramientas de software libre como Syslog para la observación, análisis y control de mensajes. Además, al aumentar la complejidad de las redes, se indica la necesidad de aplicar herramientas para robustecer los sistemas de detección de anomalías.

Como se mencionó en [31], las redes aumentan su complejidad de manera progresiva, por lo que es necesaria la aplicación de *frameworks* con mayor robustez para la detección de posibles ataques o anomalías. Por ello, en [32] se presenta el diseño e implementación de un *framework* de gestión centralizada de *logs* de aplicaciones mediante el uso de ELK. Esta herramienta permitió el procesamiento y transporte de *logs* a través de Logstash, el almacenamiento y búsqueda de eventos por medio de Elasticsearch y finalmente la visualización y exploración a través de Kibana. Se conoció las ventajas de esta herramienta, la cual dispone de una interfaz de consulta gráfica y permite la escalabilidad de la solución en relación al rendimiento de la red.

Por otro lado, en [33], [34] y [35] sugieren la aplicación de técnicas de ML para la detección de ataques en el tráfico de red. En [33] se realizó una comparación de diversos algoritmos de ML con la finalidad de conocer su capacidad para detectar comportamientos anómalos en la práctica de redes. La evaluación se realizó sobre conjuntos de datos disponibles públicamente: CICIDS-2017 y UNSW-NB15; verificando que el algoritmo Random Forest logra el mejor rendimiento en términos de precisión. De manera similar, en [35] se realiza la aplicación del algoritmo de aprendizaje supervisado AODE sobre el conjunto de datos UNSW-NB15, con la finalidad establecer un sistema de detección de anomalías que permita detectar los posibles ataques en sistemas centralizados. Este algoritmo entregó una precisión de detección del 97,26 % y un tiempo de ejecución de 7 segundos, lo cual indica su excelente desempeño. Finalmente, en [34] se efectuó una comparación entre las técnicas de aprendizaje automático y árbol de decisión, obteniendo porcentajes de precisión máxima comprendidos entre un 92.98 % y un 92.3 % respectivamente. Cabe recalcar que esta comparación se realizó mediante la aplicación de las técnicas antes mencionadas en el conjunto de datos NSL-KDD, que resuelve problemas de redundancias y datos duplicados. Continuando por esta línea, en [36] se presenta la implementación de técnicas de ML para la detección de anomalías dentro de un *dataset* basados en *Network Intrusion Detection System (NIDS)*, con la finalidad de automatizar procesos. Se aplicaron pruebas con algoritmos de aprendizaje Supervisado y No Supervisado, junto con técnicas de reducción de dimensionalidad (representación en menor dimensión de un conjunto de datos de tal manera que los datos no sufran modificaciones), obteniendo un nivel de detección de anomalías en la red del 96,06 %. Ante los altos porcentajes de fiabilidad obtenidos por parte de las técnicas de ML mencionadas, Fidalgo [37] diseñó un analista virtual mediante técnicas de ML, generando un *Intrusion Detection System (IDS)* que detecte con precisión la actividad anómala que ocurre en una red. Para ello se utilizó técnicas de aprendizaje supervisado y no supervisado, como K-Means o Random Forests, obteniendo resultados por encima del 90 % lo cual indica la fiabilidad de los algoritmos.

Como se visualiza en los documentos antes mencionados, la seguridad es un punto de importancia al momento de crear *frameworks* que fortalecen la privacidad de los datos. Las herramientas y algoritmos mencionados en los experimentos anteriores, entregaron resultados con un porcentaje alto de detección anomalías dentro de una red; sin embargo, es importante recalcar que la mayoría de los experimentos realizados en este campo son aplicados en *datasets* de dominio público que generalmente disminuyen problemas de redundancia o datos duplicados. Esto influye al momento de aplicar los algoritmos en eventos reales, ya que puede variar el porcentaje de exactitud de detección de anomalías.



Diseño e implementación del framework

4.1. Diseño del *framework*

A continuación, se describe la arquitectura propuesta para la recolección, preprocesamiento e indexación de datos mediante la pila [ELK](#). Posteriormente se plantea los procedimientos y la aplicación de las técnicas de [ML](#), con la finalidad de detectar posibles anomalías en el comportamiento del tráfico de red.

En la Figura 4.1 se aprecia la arquitectura general del *framework*. Los *logs* extraídos del [Wireless LAN Controller \(WLC\)](#) Cisco [WLAN](#) son sometidos a las siguientes etapas:

- **Recolección de datos:** En esta etapa se produce la recolección de datos del [WLC](#) perteneciente a la [IES](#) seleccionada, siendo condensados en un único archivo con formato `.csv`.
- **Pre-procesamiento:** El pre-procesamiento es realizado mediante las herramientas `Awk` y `Logstash`. Esta etapa es la encargada de la identificación de la estructura general y los tipos *logs* obtenidos en la recolección de datos. A continuación, se produce la separación y creación de nuevos archivos para cada uno de los tipos de *logs*, procesándolos para la identificación de sus variables y colocándolos en un único formato delimitado por el caracter `;`. Esto facilita su posterior indexación y almacenamiento en la herramienta `Elasticsearch`.
- **Indexación y almacenamiento:** En esta etapa se crean los índices. Estos permiten el almacenamiento de los datos preprocesados en la herramienta `Elasticsearch` para su posterior monitoreo y análisis. Los `csv` se envían a `Elasticsearch` usando *pipelines* los cuales se configuran a través de archivos.
- **Visualización de datos:** Esta etapa propone el análisis de los datos mediante la creación de *dashboards*. Estos posibilitan la realización del seguimiento del estado de las variables de cada tipo de *log* identificado y facilita la toma de decisiones.
- **Aplicación de técnicas de [ML](#):** Una vez seleccionados los *logs* a analizar, se aplica la técnica de [ML](#) apropiada mediante `Scikit Learn` y `Weka`. Como resultado se obtiene las posibles anomalías.
- **Visualización de resultados:** Finalizada la detección de anomalías, se presentan los resultados

obtenidos usando Kibana.

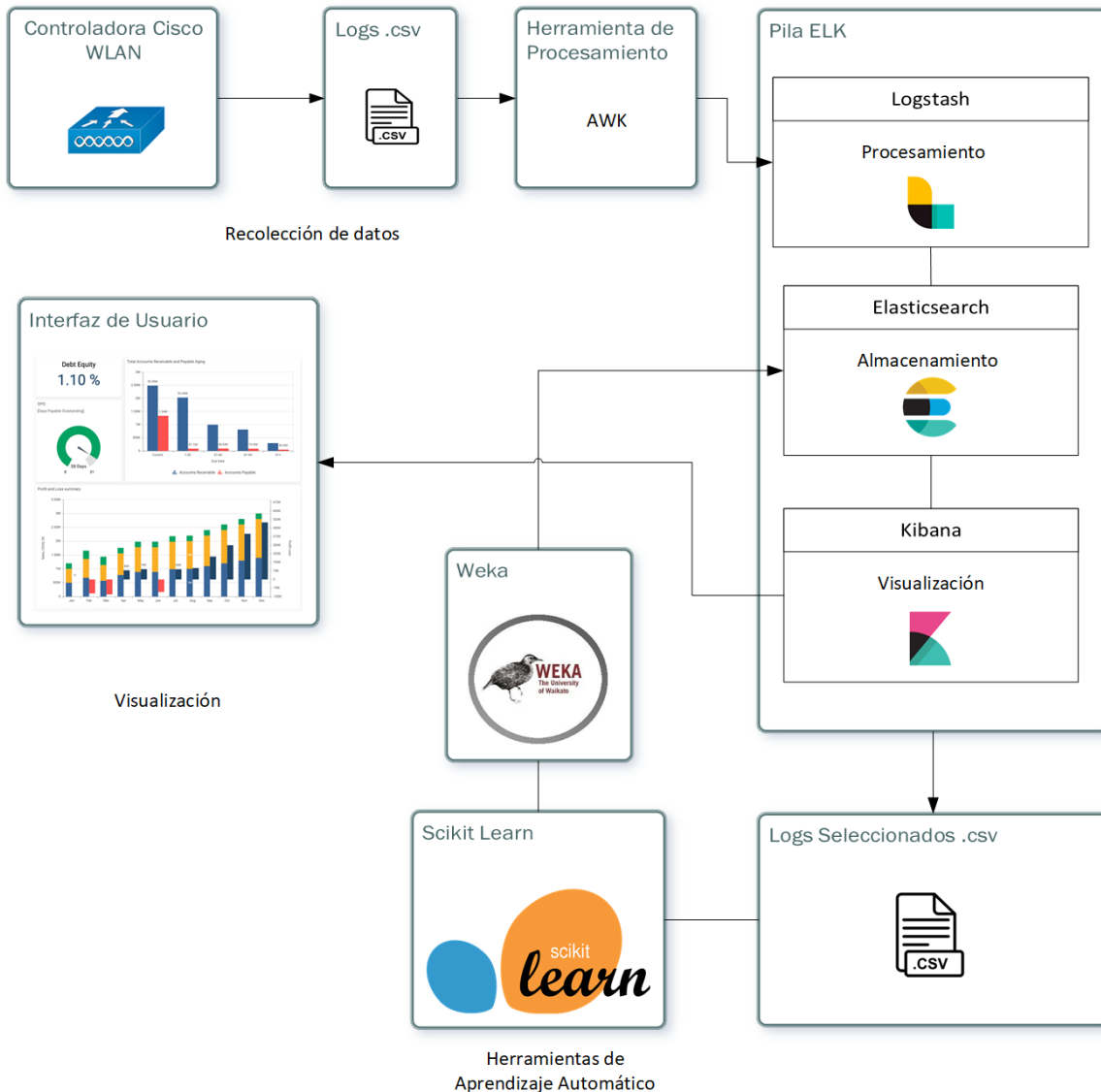


Figura 4.1: Arquitectura general del *framework*.

4.2. Implementación del sistema

A continuación, se presenta la arquitectura de servidores para la implementación de los componentes de la pila **ELK**. Para su aplicación, se propone utilizar tres máquinas virtuales, debido a que las herramientas necesitan altos recursos tanto de memoria, disco duro y procesador. Las características de las máquinas virtuales varían según los requerimientos para cada componente, siendo:

- Servidor recolector: Presenta una memoria *Random Access Memory (RAM)* de 4 Gigabytes, procesador de 1 núcleo y disco duro de 100 Gigabytes. Con las características mencionadas

anteriormente, este servidor aloja el *pipeline* de procesamiento Logstash, el cual permite la ingesta y preprocesamiento de datos.

- Servidor de índice de datos: Utiliza una memoria RAM de 8 Gigabytes, procesador de 2 núcleos y disco duro de 200 Gigabytes, permitiendo así la implementación de la herramienta de indexación Elasticsearch.
- Servidor de reporte: La infraestructura asignada para este servidor es una memoria RAM de 4 Gigabytes, un procesador de 1 núcleo y un disco duro de 50 Gigabytes, lo cual es suficiente para la instalación de la interfaz de usuario Kibana.

El sistema operativo implementado en cada una de las máquinas virtuales es Ubuntu Server 20.04.1 LTS debido a las ventajas que este sistema operativo presenta, como estabilidad, resistencia e incluso una mayor seguridad [38].

En el Anexo A se encuentran los prerequisites necesarios y proceso de instalación para cada una de las herramientas mencionadas anteriormente.

4.2.1. Preprocesamiento del *Dataset*

Un paso fundamental dentro de ML es el preprocesamiento. En este paso se realiza la limpieza, integración, transformación y simplificación de los datos mediante la herramienta Awk y Logstash, preparándose así para la etapa de indexación y almacenamiento. En esta etapa se desarrolla el preprocesamiento del documento csv que contiene la recopilación de los *logs* correspondientes al tráfico de la red inalámbrica de la IES seleccionada en un lapso de tiempo. En esta etapa se transforma el formato de la fecha de los *logs* al formato de Elasticsearch para permitir las consultas sobre fechas o manipulación de los datos en una línea de tiempo [39].

1. **Filtración de tipos de *logs*:** En esta etapa se separan los tipos de *logs* en archivos individuales para simplificar el proceso de análisis de variables.
2. **Identificación y separación de variables:** Los archivos del proceso anterior son transformados a un formato csv. En donde cada parámetro se encuentra en una columna. El delimitador del archivo csv es el caracter ";". Este archivo csv es posteriormente adicionado a Elasticsearch. El proceso de identificación de los valores de cada tipo de variable se encuentra dado por la utilización de las palabras claves seleccionadas, siendo generalmente los nombres de los atributos.
3. **Pipelines de ingesta de datos:** Al poseer los archivos individuales correspondientes a cada tipo de *log* con un formato uniforme, se desarrolla la canalización de estos mediante *pipelines*. Estos permiten la indexación y almacenamiento en Elasticsearch. La estructura de un *pipeline* es la siguiente:

input: Permite la generación de eventos mediante la ingesta de datos. En este caso puntual, la ingesta se realiza mediante el *path* debido a que se manejan archivos csv. Dentro de ella se declaran parámetros como la posición de inicio de lectura y la deshabilitación del registro de posición actual.

filter: El filtrado es la etapa intermedia, encargada de aplicar procesos preconfigurados como: declaración de las columnas presentes, el tipo de separador utilizado, formato de fecha manejado y declaración de variables eliminadas.

output: El *plugin* de salida permite la canalización de los datos a una ruta específica. En esta etapa se realiza el proceso de declaración del índice a utilizar y la dirección del *host*

correspondiente a Elasticsearch.

Al poseer archivos que manejan diferentes tipos de *logs*, se crea dentro del servidor correspondiente a Logstash un *pipeline* para cada uno de ellos.

4.2.2. Proceso de creación de índices e indexación de datos

Esta etapa presenta la indexación y almacenamiento mediante la herramienta Elasticsearch, que basa su funcionamiento en índices y patrones de índice. La creación de los índices se realiza en el servidor correspondiente al índice de datos. Para crear el índice se utiliza la sentencia mostrada en el Listado 4.1.

```
# curl -X PUT "localhost:9200/nombredelindice?pretty "
```

Listado 4.1: Sentencia de creación de índices.

En la sentencia se indica la dirección IP correspondiente al servidor que aloja Elasticsearch (*localhost*), seguido del puerto utilizado para la comunicación con el resto de servidores de la arquitectura (por defecto es 9200), y finalmente, el nombre asignado al índice. A continuación, se configura el patrón de índice dentro de la herramienta Kibana. Este permite subir los datos en su índice correspondiente y la elección del *@timestamp* para la visualización de datos en la línea de tiempo.

4.2.3. Análisis de datos

El objetivo de esta etapa es la visualización, por parte del analista de datos, de las variables existentes en cada tipo de *log* mediante *dashboards*. Esto permite que se seleccionen los *logs* para un futuro análisis según las necesidades de la *IES*.

Para este proceso se utiliza la herramienta *Lens* disponible en Kibana. En el *dashboard* se analizan los tipos de *logs* identificados, verificando el estado de las variables que los conforman y seleccionando los adecuados para la posterior aplicación de técnicas de ML.

4.2.4. Aplicación de técnicas de ML

Como se mencionó en 2.3, existen diferentes tipos de aprendizaje dentro del campo del ML; estos entregan resultados óptimos en función del entorno del análisis. En este caso específico no se poseen datos previos para un posible entrenamiento del sistema, por lo que, según 2.3.3 los algoritmos pertenecientes al aprendizaje no supervisado poseen mayor probabilidad de entregar resultados congruentes. Dentro de este tipo de aprendizaje, se sitúan los algoritmos de agrupación (clusterización) y asociación. Debido al gran volumen de datos manejado y la relación necesaria entre estos para la detección de posibles anomalías, coloca a los algoritmos de clusterización como opción apropiada, específicamente el algoritmo de K-Means. Esto es debido a sus ventajas con el manejo de grandes conjuntos de datos, su agrupación mediante la identificación de similitudes entre vectores y a los altos porcentajes de detección de anomalías presentados en experimentos como los mencionados en [37]. El proceso seguido por este algoritmo incluye etapas como la normalización de datos y la definición del número de clústers óptimo previo a la aplicación del método. Esta última etapa puede ser desarrollada mediante diferentes

algoritmos; sin embargo, debido a la simplicidad para su ejecución se seleccionó el método de Elbow (Ver 2.3.3).

1. **Normalización de datos:** La normalización de datos permite comprimir o extender los valores de las variables del *dataset* para que se encuentren en un rango definido. Esta técnica generalmente se utiliza como preparación previa a la aplicación de una técnica de ML para permitir el modelado. En este procedimiento, se tomaron los valores de cada uno de los atributos presentes en los archivos *csv*, asignándoles valores comprendidos entre 0 y 1.
2. **Aplicación del método de Elbow:** Para conocer el número de clústers óptimo para su utilización en el algoritmo de K-Means, se aplica el método de Elbow debido a su simplicidad. Las etapas definidas para este proceso son:
 - a) Carga del conjunto de datos: Permite la lectura del documento de datos normalizados.
 - b) Método de Elbow: Genera la gráfica mediante *sklearn.cluster* de la herramienta Scikit-Learn, realizando iteraciones dentro de un rango definido y siguiendo el procedimiento mencionado en 2.3.3.
 - c) Graficación: Permite la visualización de la curva correspondiente en el rango anteriormente asignado.

Para obtener el número de clústers se analiza la gráfica obtenida. De manera general, la suma de las distancias disminuye conforme aumenta el número de clústers (Ver 2.3.3). Por lo que, el número óptimo se presentará en el punto de la gráfica donde se forme un codo que implique un cambio en el valor de la suma de las distancias (valor cercano a cero).

3. **Aplicación del algoritmo de K-Means:** Una vez obtenido el número de clústers óptimo, se procedió a la aplicación del algoritmo de K-Means mediante la herramienta Weka. Esta herramienta basa el funcionamiento del algoritmo de K-Means en la asignación del valor al clúster más cercano, conforme a la distancia entre el sujeto y el centroide del clúster. El proceso se repite de manera secuencial, hasta que todos los valores se mantengan en un mismo centroide.

4.2.5. Visualización de resultados

En esta etapa se presentan los resultados obtenidos mediante la generación de *dashboards* en Kibana; en concreto, a través de la utilización de la herramienta *Lens* que forma parte de su interfaz de usuario. Esta permite la graficación de resultados, con la finalidad de que el analista de datos correspondiente a la IES, verifique las anomalías detectadas en la clusterización, a través de la creación de gráficas que relacionen las variables, ya sea entre ellas o en la línea de tiempo.



CAPÍTULO

5

Evaluación del *framework*

El objetivo del presente capítulo, es la aplicación del *framework* descrito en la Sección 4 al entorno de una IES. El entorno seleccionado para la implementación fue la Universidad de Cuenca; en este ambiente se trabajó con los *logs* del tráfico de la red inalámbrica. Esta IES posee una infraestructura de red formada por dos WLC, correspondientes al campus central y a los campus externos. Se realizaron una serie de pruebas iniciales con un primer *dataset* facilitado por esta institución; sin embargo, los resultados no fueron congruentes debido a que este no poseía un número suficiente de *logs*. Se solicitó a la Dirección de Tecnologías de Información y Comunicación (DTIC) de la Universidad de Cuenca el acceso a los datos en tiempo real para completar los datos; no obstante, desde esta dirección se indicó que debido a una expiración de licencias de los WLC, los datos no podían ser extraídos. Ante esta situación se realizó una petición formal a la UNACH, mediante el Grupo de Trabajo de Ciberseguridad de la Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia (CEDIA), solicitando los *logs* correspondientes al tráfico de la red inalámbrica en un lapso de tiempo representativo.

Una vez que la solicitud fue aprobada por parte de la dirección de tecnología de la UNACH, se procedió a analizar este escenario. Este proceso se inicia con una fase de análisis de la topología de red, con la finalidad de conocer el entorno de los dispositivos que proveen los eventos. Posteriormente, se realizó el estudio del *dataset* facilitado, identificando los diferentes tipos de eventos manejados y su comportamiento en la red; seleccionando así los de mayor interés para su estudio. Finalmente, se implementó las etapas relacionadas a la aplicación del algoritmo seleccionado mediante un estudio previo, permitiendo así la obtención de resultados y presentándolos de manera gráfica mediante *dashboards* en Kibana.

5.1. Definición del escenario

La Figura 5.1 muestra la topología de la red de la UNACH. En esta topología, el tráfico proveniente de Internet es manejado por dos enrutadores, uno con una capacidad de 10 GB perteneciente a la Red CEDIA y uno con una capacidad de 1 GB perteneciente a la empresa Telconet. Estos enrutadores

dirigen la información generada a un *firewall* empresarial con la finalidad de obtener una gestión unificada de amenazas. Este *firewall* presenta una conexión con dos *switch core*, los cuales poseen diferente capacidad de conmutación debido a la conexión establecida con el controlador CISCO WLAN, conocido como WLC, y el VMWARE.

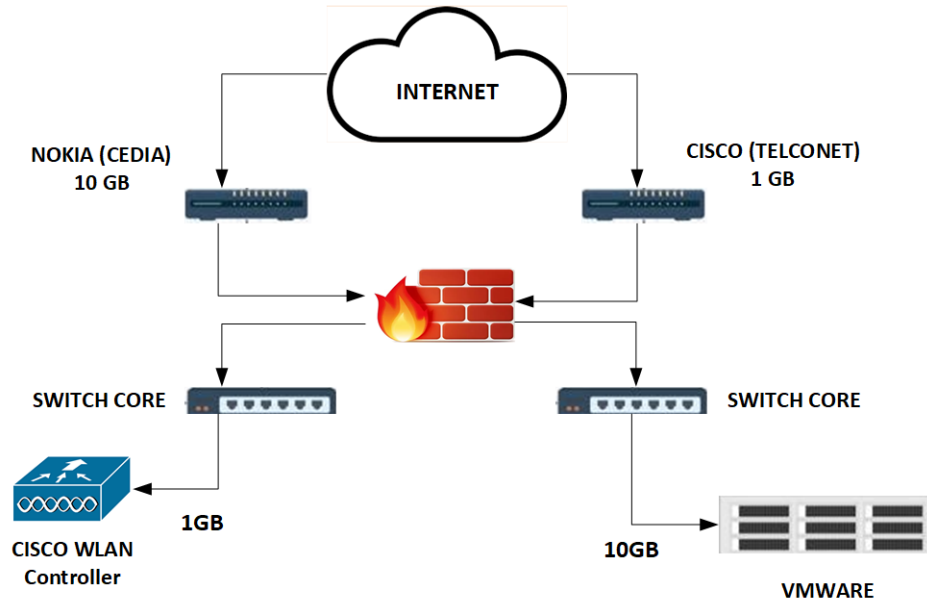


Figura 5.1: Topología de la red.

5.2. Dataset

El *dataset* manejado por el WLC corresponde al tráfico producido al comunicarse un cliente inalámbrico con un AP, por lo que los eventos capturados poseen la estructura dada por los diferentes planos identificados en una red, siendo: *management*, *control* y *data* (Ver Figura 5.2). Además, se identificó eventos de tipo *Rogue AP*, *IDS*, *RADIUS* y *Warning*.

5.2.1. Management Frames

Este tipo de evento registra la administración de la conexión inalámbrica entre el dispositivo del cliente y un AP. En la Tabla 5.1, se aprecian los distintos tipos de datos que se pueden tratar [40]. Tanto el AP como el cliente pueden enviar *frames* de deautenticación o disociación para finalizar la conexión inalámbrica; los *frames* poseen un código que indica la razón de la deautenticación. En la Tabla 5.2 se muestra los códigos de razón de deautenticación o disociación y su significado, mientras que la Figura 5.3 se visualiza la estructura de este tipo de eventos. A continuación, se listan los campos identificados en este tipo de eventos:

- Client Authenticated: MACAddress, BaseRadioMAC, Slot, UserName, IpAddr, SSID.
- Client Deauthenticated: MACAddress, BaseRadioMAC, Slot, UserName, IpAddr, Reason, ReasonCode.
- Client Association: ClientMAC, BaseRadioMAC, Slot, UserName, IPAddr.

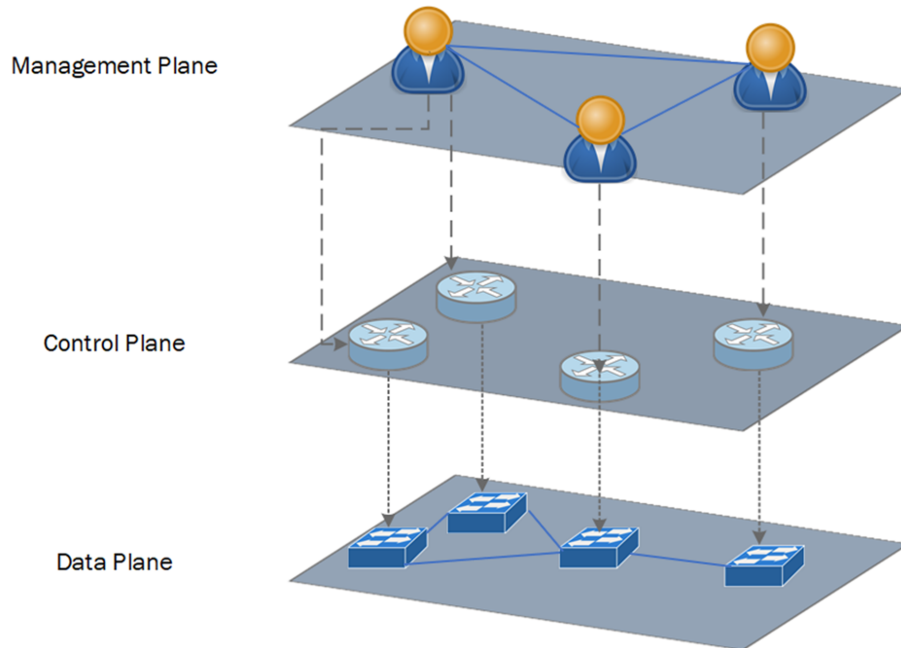


Figura 5.2: Planos de red.

- Client Disassociation: MACAddress, BaseRadioMAC, Slot, UserName, IpAddr, Reason, ReasonCode, TxPkts, TxBytes, RxPkts, RxBytes.

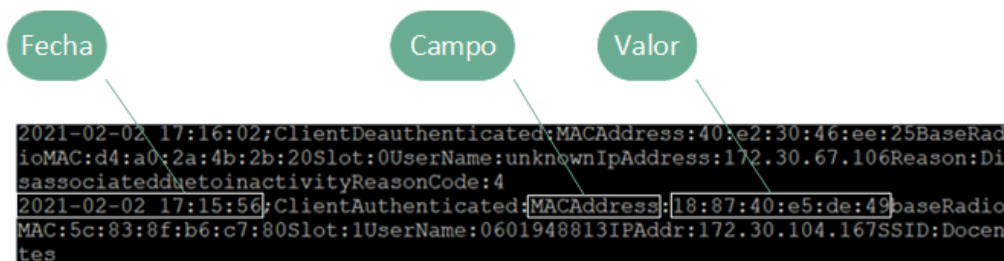


Figura 5.3: Estructura de los eventos de tipo Client Authenticated/Deauthenticated.

5.2.2. Control y data Frames

La función del plano de control es definir la lógica de control de la red utilizando diferentes protocolos; por lo tanto, estos eventos registran el reconocimiento, *Request to Send* (RTS), *Clear to send* (CTS) y ahorro de energía.

5.2.3. Rogue AP

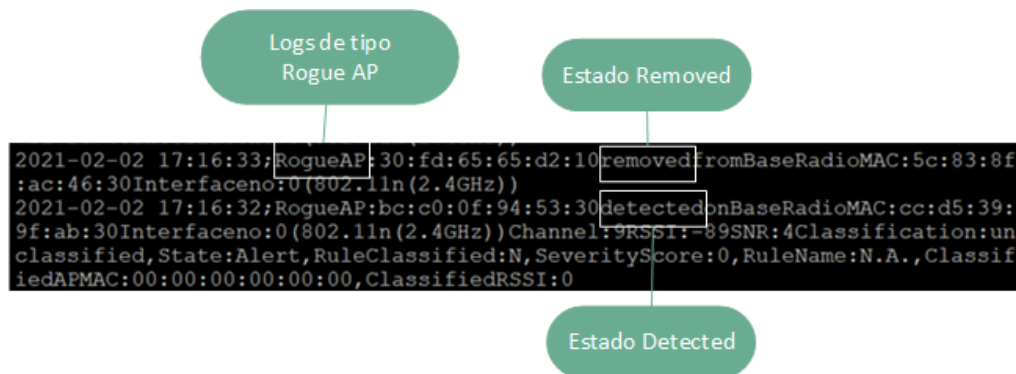
Este tipo de evento registra AP no autorizados. La estructura que presentan los eventos de tipo “Rogue AP”¹ permite dividirlo en dos tipos: removed o detected. En la Figura 5.4 se pueden apreciar los campos de cada tipo. Estos campos son:

¹ Un Rogue AP se define como un AP inalámbrico no autorizado [41]. También conocidos como “Fake AP”.

Tabla 5.1: Tipos de eventos manejados en la administración de la conexión inalámbrica.

<i>Management Frame</i>	<i>Descripción</i>
Solicitud de autenticación	Enviada por un cliente cuando desea conectarse a una WLAN .
Respuesta de Autenticación	Enviada por el AP en respuesta a la solicitud de autenticación del cliente.
Solicitud de Asociación	Posterior a la autenticación, el cliente envía una solicitud de asociación a la WLAN .
Respuesta de Asociación	Enviada por el AP en respuesta a la solicitud enviada por el cliente.
Solicitud de Reasociación	Utilizada cuando el cliente está en itinerancia.
Respuesta de Reasociación	Enviada por el AP en respuesta a la solicitud de reasociación del cliente.
Deautenticación	Enviado por el AP o el cliente para finalizar la relación de autenticación.
Disociación	Enviado por el AP o el cliente para finalizar la relación de asociación.

- Tipo removed: MACRogueAP, BaseRadioMAC,Interface No.
- Tipo detected: MACRogueAP, BaseRadioMAC,InterfaceNo, Channel, RSSI, SNR, Classification, State, RuleClassified, SeverityScore, RuleName, ClassifiedAPMAC, Classified RSSI.


Figura 5.4: Tipos de eventos *Rogue AP*.

5.2.4. IDS

[IDS](#) es un dispositivo o aplicación encargada de la detección de accesos no autorizados [\[42\]](#) o actividad maliciosa dentro de un dispositivo o red [\[43\]](#).

Los parámetros presentes en este tipo de evento dentro del *dataset* son: SignatureType, Name, Description, Track, DetectingAPName, RadioType, Preced, Hits, Channel, srcMAC.

Tabla 5.2: Códigos de deautenticación.

Código de razón de deautenticación o disociación	Significado
0	Reservado.
1	Razón no especificada.
2	Autenticación anterior no válida.
3	Desautenticado porque la estación emisora está abandonando (o se ha ido) del <i>Independent Basic Service Set (IBSS)</i> o <i>Extended Service Set (ESS)</i> .
4	Disasociado por inactividad.
5	Disasociado porque el AP no puede manejar todas las estaciones asociadas.
6	Trama de clase 2 recibida de una estación que no se encuentra autenticada.
7	Trama de clase 3 recibida de una estación que no se encuentra asociada.
8	Desasociado porque la estación emisora se va (o se ha ido) de IBSS.
9	La estación que solicita asociación no se encuentra autenticada con la estación que le corresponde.

5.2.5. RADIUS

RADIUS es un protocolo de red que ofrece mecanismos de seguridad, flexibilidad, expansión y administración de las credenciales de acceso a un recurso de red [44]. Los eventos generados por este tipo de protocolo presentan dos tipos, los cuales son: RADIUSserver y RADIUSACCTServer. Los parámetros que manejan cada una de sus estructuras son:

- RADIUSserver: IPAddress, Request, ClientMAC, User.
- RADIUSACCTServer: IPAdress, Status, Position.

5.2.6. Warning

Estos eventos son de tipo informativo, ya que generan advertencias. Su estructura está dada por: APwithBaseRadioMAC, IPAddress y MachineMACAddress.

5.3. Preprocesamiento del *dataset*

Los datos facilitados por la **UNACH** corresponden a un lapso de tiempo específico, concretamente del mes de febrero y abril del año 2021. Se analizó su estructura de manera manual, la cual se presenta en la Figura 5.5. El primer campo es la fecha y hora correspondiente al *log* con un formato [D\`ia de la semana] Mes [D\`ia del mes] HH:mm:ss A\~no. El segundo campo contiene la información capturada, generalmente se tiene direcciones *Media Access Control (MAC)*, *Internet Protocol (IP)*, interfaces o canales utilizados para la transmisión de datos, clasificaciones, estados, razón de desconexión, entre otros. Estos campos varían según el tipo de evento capturado, presentando las estructuras visualizadas en 5.2.


```
Tue Feb 2 17:16:15 2021;Client Authenticated: MAC Address:bc:7f:a4:17:df:79 b
ase Radio MAC:d4:a0:2a:4b:53:e0 Slot: 0 User Name:0602359705 IP Addr:172.30.1
04.158 SSID:Docentes
Tue Feb 2 17:16:15 2021;Client Association: Client MAC:bc:7f:a4:17:df:79 Base
Radio MAC :d4:a0:2a:4b:53:e0 Slot: 0 User Name:0602359705 IP Addr: 172.30.10
4.158
Tue Feb 2 17:16:15 2021;Client Association: Client MAC:bc:7f:a4:17:df:79 Base
Radio MAC :d4:a0:2a:4b:53:e0 Slot: 0 User Name:0602359705 IP Addr: 172.30.10
4.158
Tue Feb 2 17:16:13 2021;Rogue AP : 26:18:1d:07:30:ba removed from Base Radio
MAC : ec:bd:1d:52:68:c0 Interface no:0(802.11n(2.4 GHz))
```

Figura 5.5: Estructura general de los *logs* capturados.

Como paso previo al preprocesamiento, se modificó el formato de la fecha a yyyy-MM-dd HH:mm:ss (Figura 5.6). Internamente estas se convierten a *Coordinated Universal Time (UTC)* y se almacenan como un número largo que representa milisegundos.

Fecha con nuevo formato

```
2021-02-02 17:16:32;ClientAuthenticated:MACAddress:de:d5:c4:d0:64:7cbaseRadio
MAC:18:8b:9d:b3:cd:00Slot:0UserName:unknownIPAddr:172.30.67.97SSID:Administra
tivos
2021-02-02 17:16:19;ClientAuthenticated:MACAddress:90:06:28:6e:a3:47baseRadio
MAC:c4:14:3c:41:fe:f0Slot:0UserName:0602359705IPAddr:172.30.104.218SSID:Docen
tes
```

Figura 5.6: Fecha con formato “yyyy-MM-dd HH:mm:ss”.

Inicialmente, se filtran los eventos según el tipo y se genera un nuevo archivo utilizando Awk. Para ello, se indica la posición en la cual se va a realizar la búsqueda (siendo el segundo campo) la visualización de toda la sentencia y finalmente el nombre asignado al nuevo archivo que se genera con el resultado de la búsqueda.

```
# awk -F';' ' $2 ~ /^Tipo de log/ {print $0}' archivo.csv > logespecifico
```

Listado 5.1: Filtro configurado para los tipos de *logs*.

A continuación, se procede a la etapa de identificación y separación de variables. Como se aprecia en la Figura 5.7, inicialmente se coloca el nombre de las columnas mediante la sentencia `print`, seguido de la identificación de palabras clave que son almacenadas en diferentes variables. Estas palabras son seleccionadas una vez que se verifica que forman parte de todas las líneas correspondientes a ese tipo de evento. Por ejemplo, la palabra `IDSSignatureattackdetected` se sitúa inmediatamente después de la fecha, lo cual ayuda a la extracción de esta. Para ello, se indica que la fecha se encuentra situada en el inicio de la sentencia y finaliza antes de la palabra `IDSSignatureattackdetected`. Este procedimiento se repite para la localización de todos los atributos correspondientes a las variables.

Finalizada la etapa de filtrado, se aplicó la arquitectura de los *pipelines* mencionada en la Sección 3 para cada uno de los archivos generados y correspondientes a cada tipo de evento.

- Input: En la Figura 5.9 se presenta la infraestructura de entrada, donde se declaró la ubicación del archivo correspondiente a cada tipo de evento y su posición de inicio.
- Filter: Dentro del bloque de filtrado se mencionó las variables presentes dentro del documento recopilatorio, el formato de fecha y la sentencia correspondiente a la indexación en la variable

Identificación de una palabra clave (Nombre del campo)

```
print "Fecha;SignatureType;Name;Description;Track;DetectingAPName;RadioType"
{
  {
    iIDS=index($0,"IDSSignatureattackdetected")
    iSignatureType=index($0,"SignatureType")
    iName=index($0,"Name")
    iDescription=index($0,"Description")
    iTrack=index($0,"Track")
    iAPName=index($0,"DetectingAPName")
    iRadioType=index($0,"RadioType")
    iPreced=index($0,"Preced")
    iHits=index($0,"Hits")
    iChannel=index($0,"Channel")
    iSrcMac=index($0,"srcMac")

    ValFecha=substr($0,0,iIDS-2)
    ValSignature=substr($0,iSignatureType+14,iName-iSignatureType-15)
    ValName=substr($0,iName+5,iDescription-iName-6)
```

Localización del valor del campo a través de la palabra clave

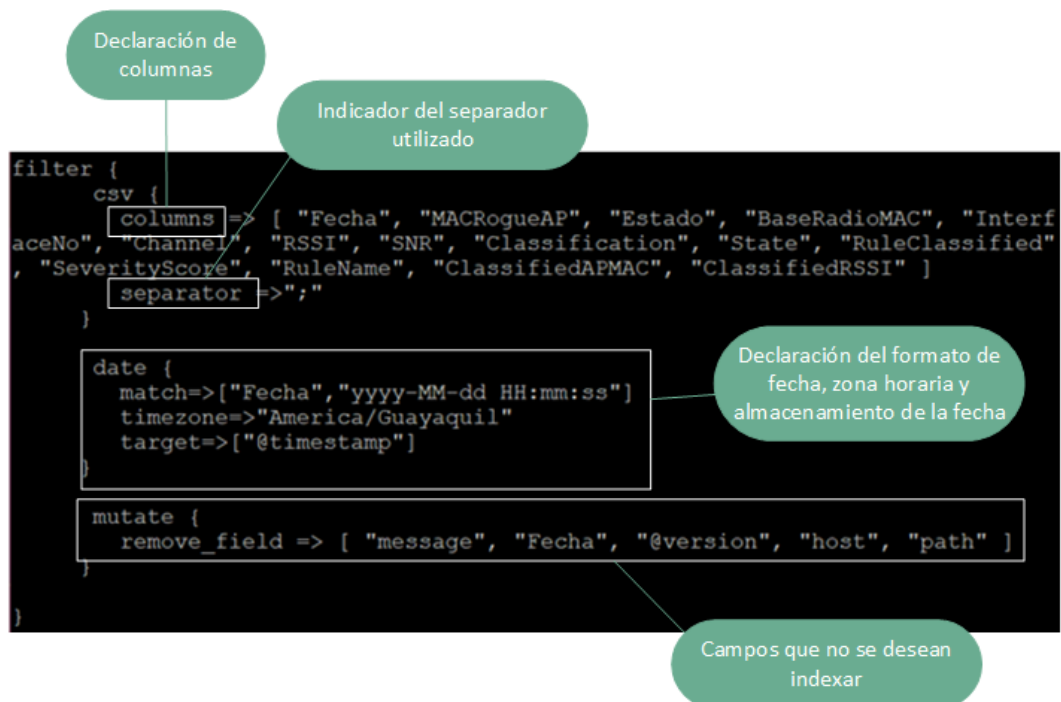
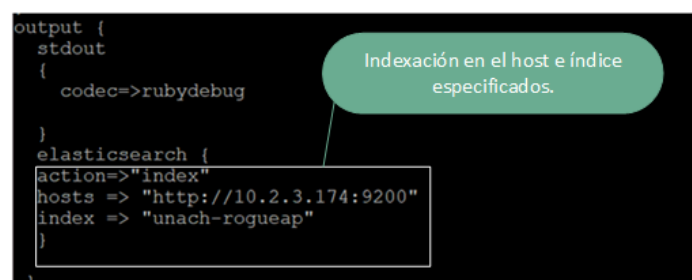
Figura 5.7: Script de identificación y separación de variables.

```
2021-02-04 07:27:06;Standard;NULLproberesp1;NULLProbeResponse-ZerolengthSSID
element;per-Mac;AP-CN-ADM-P3-DEAC;802.11a;2;1;48;DE:CB:AC:96:2C:9E
2021-02-04 07:26:39;Standard;Deauthflood;Deauthenticationflood;per-signature
;AP-CN-CTE-P3-LAB8;802.11b/g;9;500;11;48;EE:0C:43:67:7C
2021-02-05 07:57:54;Standard;Deauthflood;Deauthenticationflood;per-signature
;AP-CN-CTE-P1-SVP;802.11b/g;9;500;11;48;EE:0C:43:67:7C
2021-02-05 07:57:54;Standard;Deauthflood;Deauthenticationflood;per-Mac;AP-CN
-CTE-P1-SVP;802.11b/g;9;300;11;48;EE:0C:43:67:7C
2021-02-05 07:41:24;Standard;Deauthflood;Deauthenticationflood;per-signature
;AP-CN-BI-PB-I100E;802.11b/g;9;500;11;48;EE:0C:43:67:7C
2021-02-05 07:41:24;Standard;Deauthflood;Deauthenticationflood;per-Mac;AP-CN
-BI-PB-I100E;802.11b/g;9;300;11;48;EE:0C:43:67:7C
```

Figura 5.8: Variables identificadas y delimitadas mediante ";".

\@timestamp dentro de Elasticsearch. Además, se indica las instancias a ser removidas del proceso de indexación mediante la sentencia mutate, tal como se visualiza en la Figura 5.10.

- Output: En la salida presentada en la Figura 5.11, se indica la ruta para la indexación de los datos en un índice específico. Este se sitúa en la dirección y puerto correspondiente a Elasticsearch, siendo 10.2.3.174.

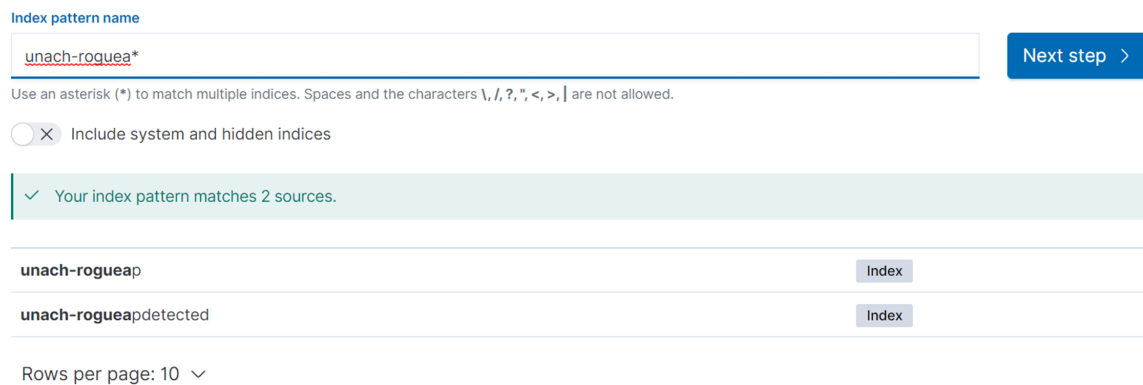
Figura 5.9: *Input* del pipeline de los eventos de tipo **Rogue AP**.Figura 5.10: *Filter* del pipeline de los eventos de tipo **Rogue AP**.Figura 5.11: *Output* del pipeline de los eventos de tipo **Rogue AP**.

5.4. Proceso de creación de índices e indexación de datos

Para esta etapa, se siguió el procedimiento descrito en la Sección 4.2.2, creando índices para cada uno de los tipos de eventos identificados. Es necesaria la creación de los patrones de índice dentro de la interfaz gráfica de Kibana, por lo que se accedió a ella mediante un proxy inverso (Nginx) detallado en el Anexo A.

Dentro de Kibana se sitúa la opción *Stack Management* que permite la creación de dichos patrones. En la Figura 5.12, se presenta la declaración del nombre del patrón, el cual debe coincidir con el nombre de índice asignado en el paso presentado en la Sección 4.2.2. Posteriormente, se selecciona *timestamp* como el campo de hora principal para usar como un filtro de hora global, el cual permitirá la visualización y manipulación de datos en la línea del tiempo.

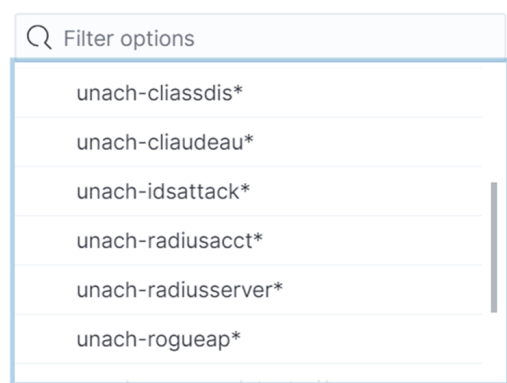
Step 1 of 2: Define an index pattern



unach-rogueap	Index
unach-rogueapdetected	Index

Figura 5.12: Creación de los patrones de índice.

Finalmente en la opción *Discover*, se visualiza los índices creados para su posterior selección y análisis (Ver Figura 5.13).



unach-cllassdis*
unach-cliaudeau*
unach-idsattack*
unach-radiusacct*
unach-radiusserver*
unach-rogueap*

Figura 5.13: Listado de índices creados.

5.5. Dashboards

En esta sección se analizaron los datos mediante la creación de *dashboards* o paneles de control usando la herramienta Kibana. Luego, se aplicaron técnicas de ML; el Anexo B presenta el procedimiento de creación. Como análisis inicial, se generó un gráfico mediante la opción *Discover* presente en la interfaz gráfica de la herramienta. Esta permite la visualización de los datos correspondientes al tipo de evento en la línea temporal, como se presenta en la Figura 5.14; indicando la fecha de inicio y finalización de la captura, número de eventos detectados y almacenados cada día. Dicha gráfica presenta un lapso en el cual no se posee información, debido a que el *dataset* facilitado por la UNACH no registró datos para este intervalo de tiempo, por lo que el análisis y posterior toma de decisiones en relación a los eventos a utilizar se vieron afectados por este lapso de tiempo desierto.

Finalmente, en la parte inferior de la gráfica presentada en la Figura 5.14, se visualiza a detalle cada uno de los eventos junto a su fecha y variables. Estas coinciden con las variables declaradas en el archivo *pipeline* generado para la indexación de datos.

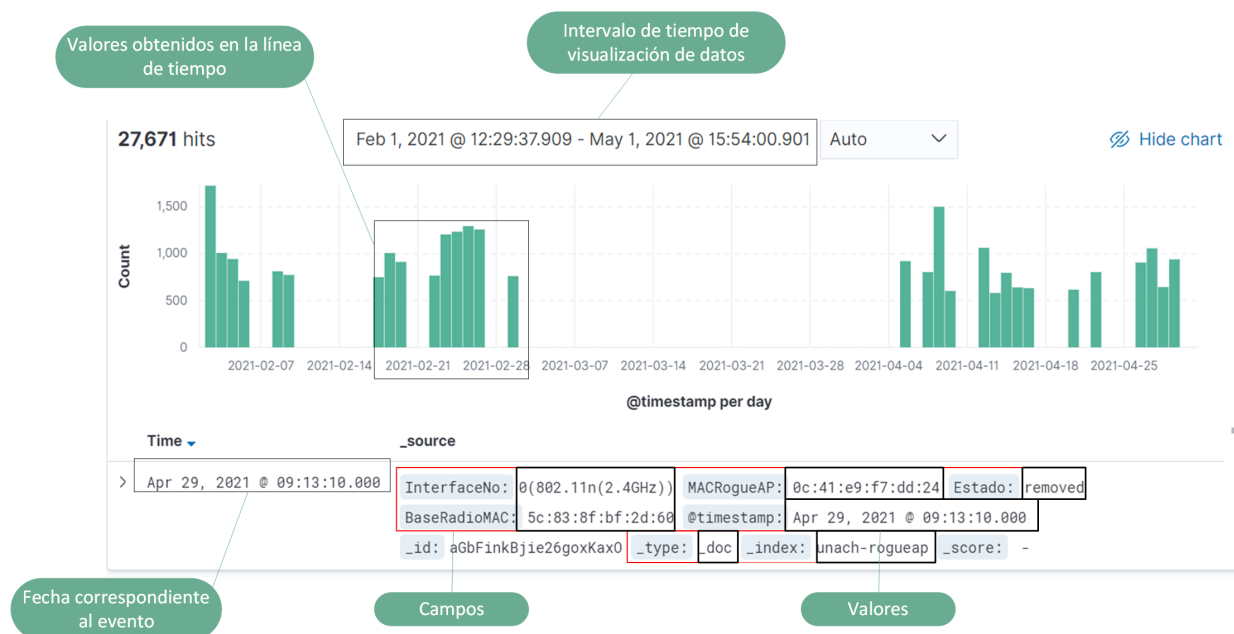


Figura 5.14: Visualización de datos en la línea temporal.

5.5.1. Análisis mediante Dashboards

En esta sección se presenta los *dashboards* generados para cada uno de los tipos de eventos identificados, permitiendo así realizar un análisis general de las variables que los conforman y seleccionando los más adecuados para la posterior aplicación de técnicas de ML.

- **Eventos de tipo Client Authenticated/Deauthenticated:** Las variables que forman los eventos de tipo Client Authenticated/Deauthenticated, presentan los estados visualizados en el Anexo C, Figura C.1. En este *dashboard* se aprecia cómo las 9 variables presentan más de un estado, lo que es de relevancia para obtener resultados más precisos a futuro. Por ejemplo, en el caso de las variables Client y Slot, únicamente poseen dos estados, mientras que el resto de variables poseen

más de dos. Una variable relevante es `UserName` (Figura 5.15), la cual indica el nombre de usuario que desea autenticarse o desautenticarse de un `AP` en concreto.

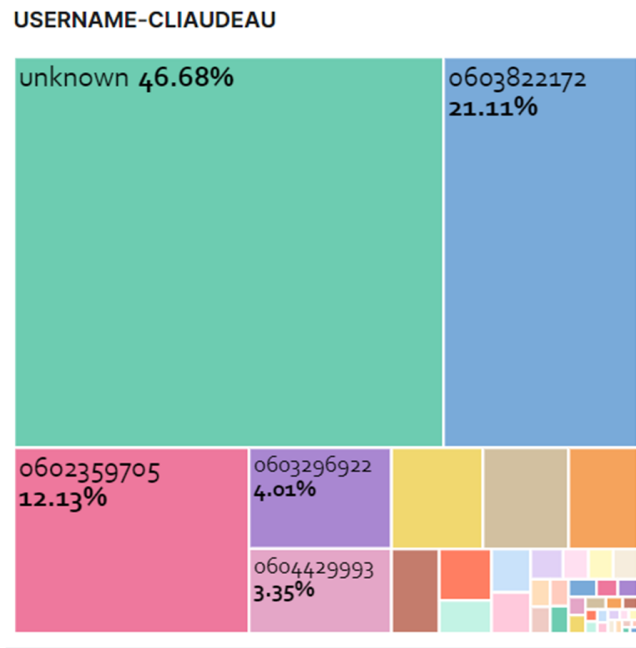


Figura 5.15: Estado de la variable `username`

La `UNACH` generalmente maneja como identificador números de cédula, nombres de usuario o correos asignados por la institución. Al realizar un análisis de estos datos, un gran porcentaje corresponde a usuarios que cumplen con estos parámetros; sin embargo, se detectaron una gran cantidad de usuarios de tipo `unknown` y en menor porcentaje el intento de autenticación por parte de un usuario `anonymous`. Estos valores no reconocidos por la institución son anomalías, las cuales podrían representar posibles intentos de accesos no autorizados por parte de terceros, lo que implica una amenaza a la seguridad.

- **Eventos de tipo `IDSAttack`:** Presentan un total de 10 variables. Todas poseen dos o más estados posibles, excepto `SignatureType` que se visualiza un único estado presentado en el Anexo C, Figura C.2. Este campo no se considera como un campo de decisión debido a que posee el mismo valor en todos los casos; este campo fue descartado. Dos de los campos más relevantes en este tipo de `logs` son `APName` y `Description`, ya que indican el `AP` que detecta el posible ataque junto a la descripción del suceso que puede implicar una amenaza a la seguridad o anomalía. Este tipo de `log` es un excelente candidato para futuros análisis a profundidad; sin embargo, el número de eventos disponibles de tipo `IDSAttack` es bajo en relación al resto de eventos identificados, por lo que los resultados que se obtendrían de una aplicación de técnicas de `ML` no serían congruentes.
- **Eventos de tipo `RogueAP`:** En la Figura C.3 se presenta el estado de los campos de los eventos de tipo `Rogue AP`, donde se identificaron 14 campos. Se visualiza que siete de ellos poseen el mismo valor en todos los casos, por lo que no devuelven información relevante al realizar un análisis; estos campos fueron descartados. Estos campos son: `ClassifiedAPMAC`, `Classification`, `ClassifiedRSSI`, `RuleName`, `RuleClassified`, `SeverityScore` y `State`. Como se mencionó anteriormente, este tipo de eventos representan `AP` no autorizados, por lo que la gráfica respectiva a los estados

removed o detected es de suma importancia para conocer los porcentajes de cada uno de ellos. En este caso, existe un 47,99 % de eventos de este tipo que han sido detectados frente a un 52.01 % que han sido removidos, indicando una posible anomalía o amenaza a la seguridad debido a la similitud de los porcentajes. De igual manera, una variable que presenta comportamientos inusuales es Channel, ya que de forma predeterminada, la mayoría de los dispositivos actuales están cerca de los 2,4 GHz, siendo los canales 1, 6 y 11 los más utilizados. En este caso, se visualiza la presencia de los canales 2 y 3 dentro de los canales más utilizados, por lo que podría implicar una posible anomalía que afectaría al tráfico de red (Figura 5.16).

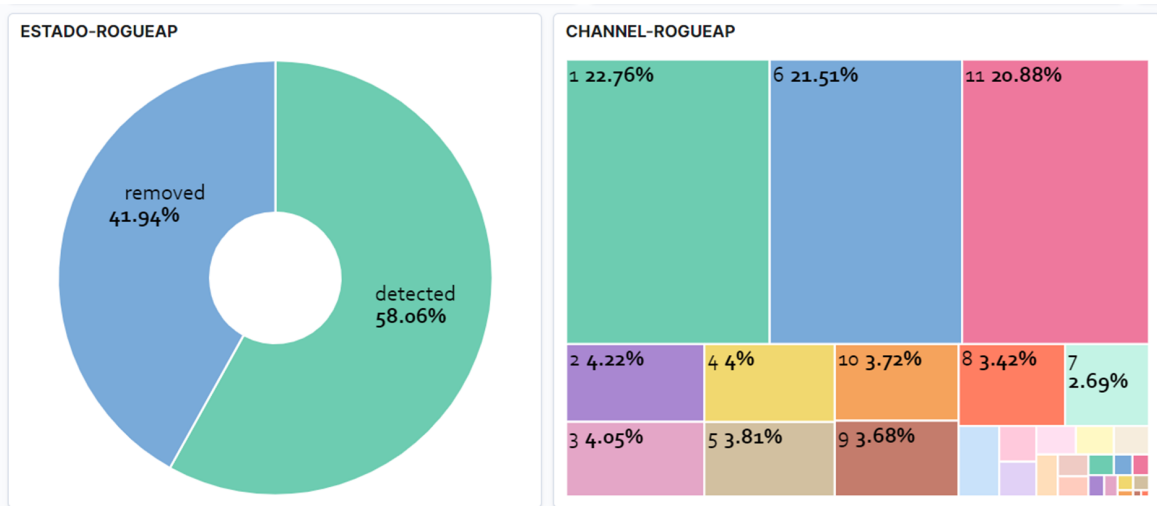


Figura 5.16: Estado de los campos Estado y Channel.

- **Eventos de tipo RADIUS:** Dentro de los *logs* de tipo RADIUS, se diferenciaron dos tipos:
 - **RADIUSACCT:** Este tipo de *log* está conformado únicamente por tres campos: IP, Status y Position. Los campos IP y Status poseen únicamente dos estados posibles, mientras que el campo Position registra un solo valor. Este tipo de *log* es descartado debido a que el número de atributos es reducido, y los valores que estos manejan no representan variables de decisión.
 - **RADIUSserver:** Los *logs* de tipo RADIUS Server presentan un total de 5 variables, de las cuales la variables Status únicamente registra un solo valor, siendo automáticamente descartada como variable de decisión. Tanto los campos IPAddress como Request, poseen un estado predominante que representa el 98.81 % y 96 % en cada uno de los casos, por lo que estos campos no se consideran como campos de decisión. Finalmente, se posee los campos ClientMAC y UserName, los cuales presentan diversos valores, tales como números de cédula o nombres de usuario asignados por la institución, siendo candidatas para futuros análisis; sin embargo, dentro del contexto de este tipo de *logs* no es posible la aplicación de técnicas de ML debido al escaso número de variables.
- **Eventos de tipo Warning:**

Este tipo de eventos están conformados únicamente por tres campos, los cuales poseen porcentajes distribuidos de manera homogénea para cada valor identificado, siendo en los campos APBaseRadioMAC y MACAddress cuatro estados posibles respectivamente y en el campo IPAddress dos estados posibles. La información entregada por este tipo de eventos es escasa, debido al

número de campos manejado y sus estados, por lo que queda descartado para un posible análisis en profundidad.

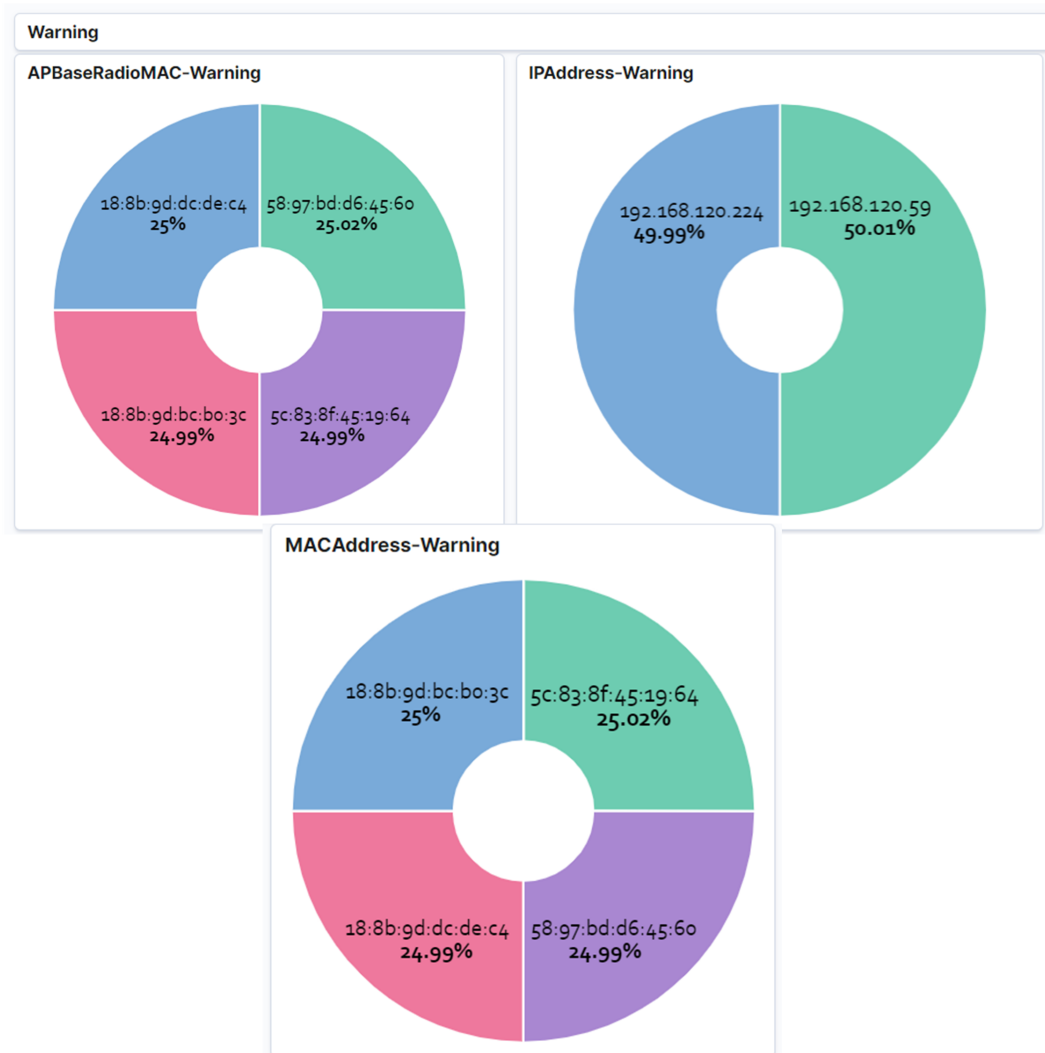


Figura 5.17: Variables manejadas por los *logs* de tipo Warning

5.6. Aplicación de técnicas de ML

Según el análisis realizado en la Sección 5.5.1, se aplicó el algoritmo de clusterización seleccionado; en este caso el algoritmo de K-Means. El procedimiento seguido para esta etapa se encuentra descrito en la Sección 4.2.4, siendo ejecutado en los siguientes casos:

5.6.1. Eventos de tipo *Rogue AP*

Como se mencionó anteriormente, los *logs* registrados de tipo *Rogue AP* son pertenecientes a *AP* no autorizados, por lo que su análisis es de vital importancia para conocer los *AP* que registran un mayor porcentaje de detección de *AP* no autorizado, además de conocer su relación con el resto de

variables manejadas. Para este *dataset* se identificaron 14 variables, de las cuales 6 fueron eliminadas al poseer un único estado y no ser una variable de decisión que afecte a la agrupación de los datos. Para este análisis, se seleccionó únicamente las filas pertenecientes al estado *detected*, ya que estos eventos no sufren alguna acción inmediata como en el caso de los eventos de tipo *removed*.

Inicialmente se llevó a cabo la normalización de datos explicada en el Sección 1, asignando valores entre 0 y 1, seguido de la aplicación del método de Elbow, obteniendo la gráfica presentada en la Figura 5.18. En esta se visualiza como la curva forma un codo e inicia su atenuación en un valor de cinco, siendo el número óptimo de clústers.

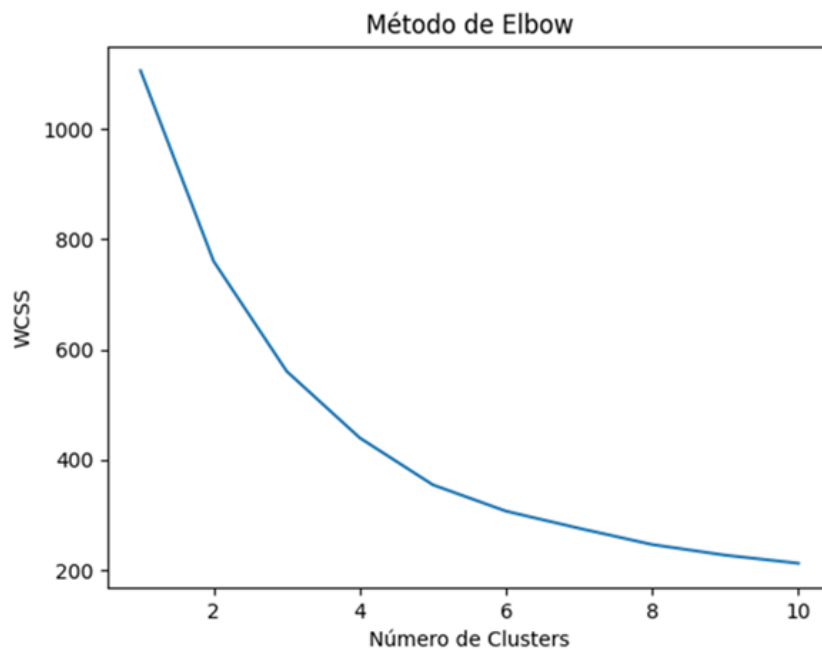


Figura 5.18: Número de clústers óptimo correspondiente a los eventos de tipo *Rogue AP*.

- **Aplicación del algoritmo para 5 clústers:** Como primer experimento se generó la aplicación del algoritmo para cinco clústers dentro de la herramienta Weka, la cual devolvió los resultados presentados en la Figura 5.19. En la parte izquierda se sitúa el nombre de los atributos junto los valores obtenidos para cada clúster, presentando los valores según la normalización realizada previamente. Como los valores no son identificables a primera vista debido a que se presentan dentro de un rango entre 0 y 1, fue necesario realizar la correspondencia a los datos sin normalizar.

1. Primer clúster: Posee una MACRogueAP 34:0a:94:17:51:63, BaseRadioMAC cc:d5:39:9f:ac:f0, InterfaceNo 0(802.11b/g), Channel 6, RSSI -89 y SNR 3.
2. Segundo clúster: Posee una MACRogueAP bc:e0:0f:94:0a:e0, BaseRadioMAC 5c:83:8f:d7:e0:f0, InterfaceNo 0(802.11n(2.4GHz)), Channel 6, RSSI -86 y SNR 5.
3. Tercer clúster: Posee una MACRogueAP b0:4e:26:23:ad:84, BaseRadioMAC 70:b3:17:7e:4f:00, InterfaceNo 0(802.11a/b/g/n), Channel 36, RSSI -89 y SNR 3.
4. Cuarto clúster: Posee una MACRogueAP 3c:e8:24:6d:96:5c, BaseRadioMAC 5c:83:8f:b6:c2:10, InterfaceNo 0(802.11n(2.4GHz)), Channel 6, RSSI -86 y SNR 4.

```

              Cluster#
Attribute    Full Data    0      1      2      3      4
              (5355.0)  (913.0) (1500.0) (1004.0) (1131.0) (807.0)
=====
MACRogueAP   0.5047   0.1953   0.7107   0.6494   0.2236   0.6861
BaseRadioMAC 0.4897   0.7438   0.2421   0.5954   0.2603   0.8524
InterfaceNo  0.1017   0.1482   0.017    0.3446   0.0188   0.0207
Channel      0.0499   0.0268   0.0252   0.1549   0.0256   0.0255
RSSI         0.3113   0.3041   0.3277   0.2648   0.3255   0.327
SNR          0.2617   0.249    0.2715   0.2499   0.2671   0.2651

Time taken to build model (full training data) : 0.05 seconds

== Model and evaluation on training set ==

Clustered Instances

0      913 ( 17%)
1     1500 ( 28%)
2     1004 ( 19%)
3     1131 ( 21%)
4      807 ( 15%)

```

Figura 5.19: Resultados obtenidos con cinco clústers para los logs de tipo *Rogue AP*.

5. Quinto clúster: Posee una MACRogueAP b8:69:f4:6a:fe:c4, BaseRadioMAC ec:bd:1d:3d:68:b0, InterfaceNo 0(802.11n(2.4GHz)), Channel 6, RSSI -86 y SNR 4.

- **Análisis de resultados:** Los resultados indican que los datos se distribuyen principalmente en los clústers 2 y 4, con un porcentaje de 28 % y 21 % correspondientemente. El segundo clúster posee una BaseRadioMAC 5c:83:8f:d7:e0:f0, siendo el que detecta un mayor número direcciones MAC relacionadas a *Rogue AP*, mientras que el cuarto posee una BaseRadioMAC 5c:83:8f:b6:c2:10. Estas coinciden con el número de interfaz 0(802.11n(2.4GHz)) y canal 6, siendo valores dentro de los rangos normales. La mayoría de dispositivos operan, por defecto, en la franja de frecuencias cercana a 2.4 GHz. Esta franja de frecuencias se divide en 14 canales que se encuentran separados por 5 MHz, existiendo un inconveniente en el problema de distribución debido a que cada canal necesita 22MHz para su operación, por lo que se pueden producir solapamientos entre canales [45]. En la UNACH se usa de manera predeterminada los canales 1, 6 y 11 para la franja de frecuencias de 2.4GHz; esto para evitar solapamientos, sin embargo, puede existir la asignación de canales diferentes de manera automática. En el tercer clúster se aprecia la utilización del canal 36, el cual pertenece a la franja de frecuencias cercana a 5 GHz. Por otro lado, los valores obtenidos en cada clúster para la relación señal-ruido SNR, se sitúan entre 3 dBm y 5 dBm, siendo valores que representan interferencia. Esto se debe a que el valor óptimo de SNR se sitúa entre 18 y 30 dBs, siendo el valor máximo la representación de la mejor calidad de señal [46]. Esto está relacionado estrechamente con el valor de RSSI obtenido, los cuales presentan valores en los clústers entre -86 y -89 dBm. Los valores obtenidos indican anomalías en las intensidades de señal y en el rendimiento, que pueden ser causados generalmente por interferencias entre los canales.

5.6.2. Eventos de tipo *Client Authenticated/Deauthenticated*

Los eventos de tipo *Client Authenticated/Deauthenticated* entregan información relacionada con la conexión entre el dispositivo del cliente y el AP. Por lo tanto, la información entregada por este tipo de eventos es relevante para la detección de posibles anomalías. Como se mencionó en la Sección 5.5.1, este tipo de eventos presentan un total de 9 campos considerados de decisión debido a que presentan más de dos estados. El 93 % de los eventos registrados corresponden a *logs* de tipo *Authenticated*. Debido a este alto porcentaje, es de relevancia conocer la relación existente entre este con el resto de campos, tales como el *UserName* o *SSID*.

De manera similar al proceso realizado en la Sección 5.6.1, se realizó la normalización de datos, y luego se aplicó el método de Elbow. Este procedimiento entregó los resultados presentados en la Figura 5.20. Se aprecia que la curva posee un codo e inicia su atenuación en un valor de cuatro clústers, lo que indica el número óptimo para su aplicación dentro del algoritmo de K-Means.

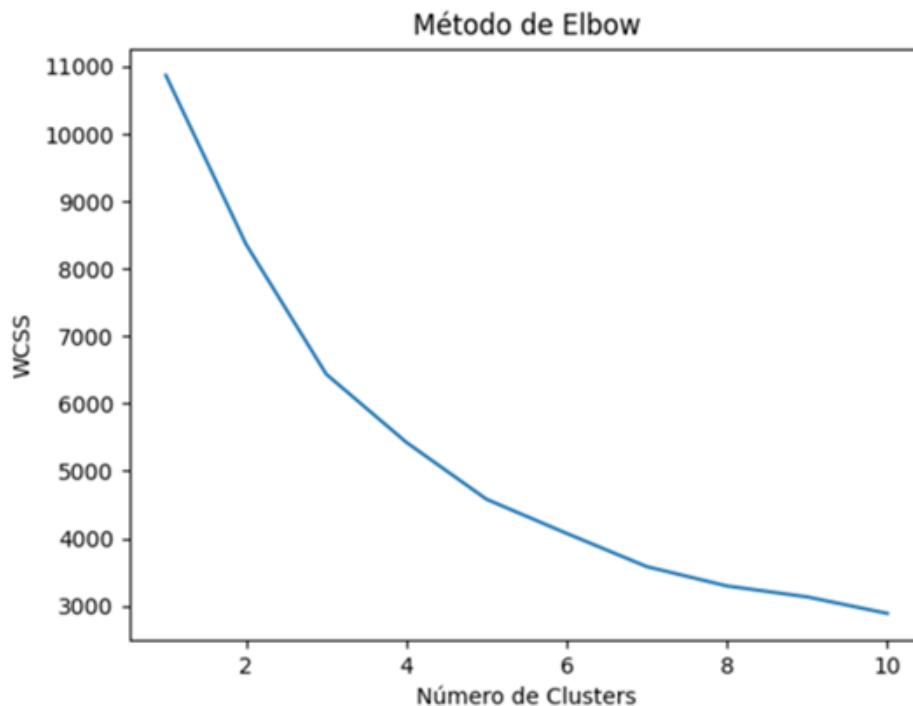


Figura 5.20: Número de clústers óptimo correspondiente a los eventos de tipo *Client Authenticated/Deauthenticated*.

- **Aplicación del algoritmo para cuatro clústers:** En la Figura 5.21 se visualizan los resultados obtenidos, siendo:
 1. Primer clúster: *Client Authenticated*, *MACAddress* 7c:23:02:93:e1:0c, *BaseRadioMAC* 5c:83:8f:dc:d0:f0, *Slot* 0, *UserName* unknown, *IpAddr* 172.30.100.104, *SSID* CEDIA, *Reason* 0, *ReasonCode* 0.
 2. Segundo clúster: *ClientAuthenticated*, *MACAddress* 88:29:9c:77:30:31, *BaseRadioMAC* 5c:83:8f:dc:d5:b0, *Slot* 0, *UserName* 603027178, *IpAddr* 172.30.105.125, *SSID* Docentes, *Reason* 0, *ReasonCode* 1.

3. Tercer clúster: ClientAuthenticated, MACAddress 88:63:df:a6:0f:65, BaseRadioMAC 5c:83:8f:dc:d0:f0, Slot 0, UserName unknown, IpAddr 192.168.181.89, SSID CEDIA, Reason 0, ReasonCode 2.
4. Cuarto clúster: ClientAuthenticated, MACAddress 88:63:df:a6:0f:65, BaseRadioMAC 5c:83:8f:c2:16:40, Slot 1, UserName anonymous@uam.es, IpAddr 172.30.108.160, SSIDCodigoTrabajo, Reason 0, ReasonCode 1.

Attribute	Cluster#				
	Full Data (10099.0)	0 (1547.0)	1 (4283.0)	2 (2256.0)	3 (2013.0)
Client	0.0679	0.0291	0.0579	0.1285	0.0512
MACAddress	0.5007	0.4511	0.5066	0.5127	0.513
BaseRadioMAC	0.4822	0.493	0.5042	0.4956	0.4124
Slot	0.1993	0	0	0	1
UserName	0.5854	0.9797	0.2239	0.9759	0.6137
IpAddr	0.399	0.0092	0.2952	0.8302	0.4361
SSID	0.3422	0.2676	0.4207	0.279	0.3035
Reason	0.0262	0.0103	0.0254	0.0408	0.0237
ReasonCode	0.0077	0.002	0.0054	0.0179	0.0057

Time taken to build model (percentage split) : 0.05 seconds

Clustered Instances

```

0      739 ( 14%)
1     2231 ( 43%)
2     1153 ( 22%)
3     1081 ( 21%)

```

Figura 5.21: Resultados obtenidos con cuatro clústers para los eventos de tipo Client Authenticated/Deauthenticated.

- **Análisis de resultados:** Como se visualiza en la Figura 5.21, el segundo clúster posee el 43 % de los *logs*, siendo clientes de tipo Authenticated con un UserName dado por un número de cédula. La UNACH utiliza generalmente como nombres de usuario los números de cédula, correos institucionales o nombres asignados, por lo que este clúster cumple con los parámetros definidos para este campo. Sin embargo, si se aprecia los resultados obtenidos para el campo UserName dentro los clústers uno, tres y cuatro, existe la presencia de usuarios identificados como unknown y *anonymous@uam.es*, representando el 57 % de los datos en total. Este tipo de *logs* pueden indicar la presencia de posibles ataques por parte de terceros no autorizados, debido a que no cumplen los estándares de la UNACH. Estos podrían indicar una potencial amenaza para el desempeño de red de la IES, ya que los clústers que identifican estos *logs* son de tipo Client Authenticated. Para los clústers uno y tres, el SSID predominante es CEDIA, lo que indica que este identificador de red es el que registra un mayor porcentaje de autenticación de usuarios de tipo unknown, seguido por el SSID Codigo Trabajo perteneciente al cuarto clúster, que registra la autenticación de usuarios de tipo anonymous. Finalmente, los cuatro clústers generados poseen un valor 0 en el campo Reason, debido a que esta variable es manejada en los eventos de tipo Client Deauthenticated

que representan un total del 6,99 % del total de eventos de este tipo, un valor poco significativo en relación al 93.01 % de clientes autenticados.

5.6.3. Relación entre UserName y SSID

En esta sección se presenta la aplicación del algoritmo de K-Means enfocado únicamente a la relación entre las variables UserName y SSID, con la finalidad de verificar su relación y comportamiento con el resto de identificadores de red existentes en la IES.

Para este tercer experimento, se siguió el procedimiento de normalización de datos para la asignación de valores comprendidos entre 0 y 1, seguido de la aplicación del método de Elbow, presentado en la gráfica de la Figura 5.22. En esta gráfica se aprecia la presencia del codo y el inicio de la atenuación de la curva entre tres y cuatro clústers, por ello se decidió realizar el experimento para estos dos valores.

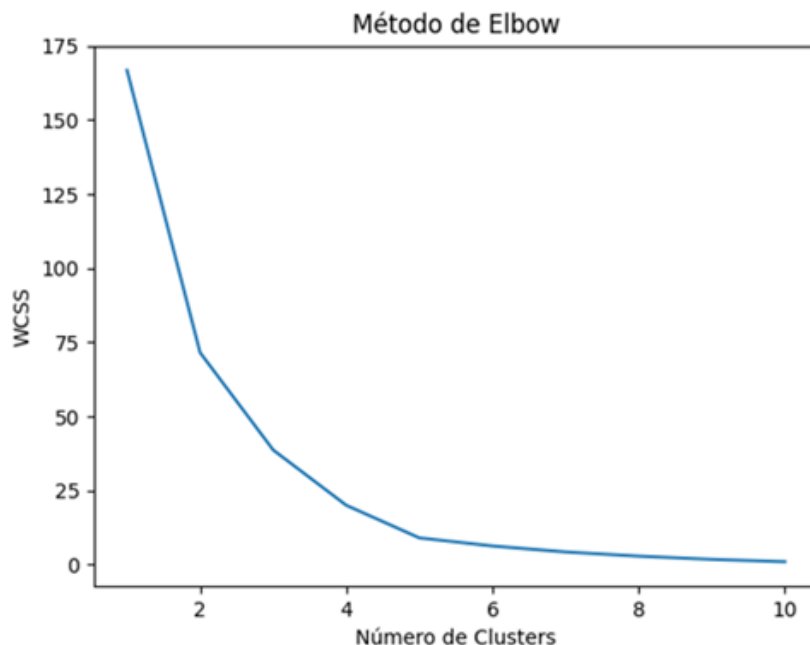


Figura 5.22: Número de clústers óptimo correspondiente a la relación entre las variables UserName y SSID.

- **Aplicación del algoritmo (Tres clústers):** En la Figura 5.23 se presenta los resultados obtenidos para tres clústers, donde:
 1. Primer clúster: UserName valeveloz98, SSID Administrativos.
 2. Segundo clúster: UserName valeveloz98, SSID Auditorio.
 3. Tercer clúster: UserName unknown y SSID CEDIA.
- **Análisis de resultados:**

Como se aprecia en la Figura 5.23, el tercer clúster identificado posee el 72 % de los datos con UserName unknown y el SSID CEDIA. Con esta clusterización se corrobora la información obtenida en 5.6.2, donde los usuarios de tipo unknown se autentican principalmente en este identificador.

```
Final cluster centroids:

Attribute      Full Data      Cluster#
                (6710.0)      0          1          2
                (1549.0)      (331.0)      (4830.0)
=====
UserName      0.9929          1          1          0.9902
SSID          0.2085          0          0.1          0.2828

Time taken to build model (full training data) : 0.05 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      1549 ( 23%)
1       331 (  5%)
2      4830 ( 72%)
```

Figura 5.23: Resultados obtenidos con tres clústers correspondientes a la relación entre las variables UserName y SSID.

Por otro lado, el primer y segundo clúster presentan un UserName que cumple con la estructura dada por la [IES](#), además de presentar la autenticación en los SSID Administrativos y Auditorio.

- **Aplicación del algoritmo (Cuatro clústers):** A continuación se aplicó el algoritmo con un total de cuatro clústers, presentando los resultados en la Figura 5.24. Los valores correspondientes a estos clústers son:
 1. Primer clúster: UserName valeveloz98 y SSID Administrativos.
 2. Segundo clúster: UserName valeveloz98 y SSID Auditorio.
 3. Tercer clúster: UserName unknown y SSID Código Trabajo.
 4. Cuarto clúster: UserName valeveloz98 y SSID [CEDIA](#).
- **Análisis de resultados:** En este caso, el tercer clúster presenta el mayor número de *logs*, manejando un UserName de tipo unknown y autenticándose al SSID Código Trabajo. Con este resultado se aprecia que los SSID con mayor autenticación de clientes que pueden representar posibles amenazas o anomalías son [CEDIA](#) y Código Trabajo, mientras que los clústers 1, 2 y 4 presentan autenticaciones a SSID por parte de usuarios autorizados.

```
Final cluster centroids:

Attribute      Full Data      Cluster#
                (6710.0)      0          1          2          3
                (1549.0) (331.0) (3196.0) (1634.0)
=====
UserName        0.9929          1          1      0.9852          1
SSID            0.2085          0          0.1      0.3251          0.2

Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      1549 ( 23%)
1       331 (  5%)
2      3196 ( 48%)
3      1634 ( 24%)
```

Figura 5.24: Resultados obtenidos con cuatro clústers correspondientes a la relación entre las variables UserName y SSID.

5.7. Visualización de resultados

Una vez detectadas las posibles anomalías mediante K-Means, se presentaron los resultados obtenidos mediante *dashboards* desarrollados en Kibana.

En la Figura 5.25 se presenta el *dashboard* correspondiente a las anomalías detectadas en los *logs* de tipo [Rogue AP](#). Estos poseen un total de 27671 *logs* con un 41,94 % en estado *removed* y un 58,06 % en estado *detected*. Al colocar esta información en una línea de tiempo, se visualiza el alto porcentaje de eventos de tipo *detected* que no sufren ningún tipo de acción inmediata, como es el caso de los tipo *removed*, lo cual implica una posible amenaza a la seguridad. Cabe recalcar que existen picos en la línea de tiempo que registran un mayor número de registros de *logs* de tipo *detected*, siendo una anomalía en el tráfico, ya que generalmente estos rondan los 600 eventos aproximadamente. El análisis continua con la generación del gráfico relacionando las BaseRadioMAC de los [AP](#) con los canales utilizados, visualizando la utilización predominante del canal 1, 6 y 11 correspondiente al rango de 2.4GHz. Sin embargo, en algunos casos se aprecia la utilización de canales como 2, 3, 4, entre otros. Debido a esto, se presentó una gráfica de barras que indique el número de *logs* registrados en cada canal, corroborando la utilización de los canales principales antes mencionados y la presencia de canales que puedan implicar posibles interferencias. Estas interferencias implican una anomalía dentro de la red, ya que puede afectar a la velocidad de transmisión de datos. Como se mencionó en la Sección 5.6.1, la banda de frecuencias más utilizada en esta institución es de 2.4GHz, siendo apoyada por la banda de frecuencias de 5GHz. Finalmente se presenta un mapa de calor donde se visualiza las BaseRadioMAC de los eventos de tipo *detected* de la [IES](#), indicando que el [AP](#) con dirección 5c:83:8f:d7:e0:f0 detecta un mayor número de [AP](#) no autorizados a lo largo de la línea del tiempo.

Por otro lado, en la Figura 5.26 se presentan gráficos relacionados con las anomalías detectadas



en los eventos de tipo *Client Authenticated/Deauthenticated*. Inicialmente se muestra el número de eventos manejados, un total de 15303, donde el 93,01 % corresponden a clientes de tipo *Authenticated*. En la gráfica correspondiente a los estados en la línea de tiempo, los eventos de tipo *Authenticated* alcanzan picos que sobrepasan los 500 eventos registrados, mientras que los de tipo *Deauthenticated* obtienen como pico máximo el registro de 80 eventos. Debido al alto porcentaje de clientes en estado *Authenticated*, se presentó un mapa de calor que relaciona las variables *UserName* y *SSID*; con la finalidad de conocer la presencia de cada tipo usuario en los diferentes *SSID*. De esta manera se verificó la presencia del *UserName unknown* principalmente en los *SSID* [CEDIA](#) y Código trabajo, resultados que coinciden con los obtenidos en la clusterización y presentan una potencial anomalía en el tráfico de red. Finalmente, se generó una gráfica correspondiente a la utilización de los *SSID* en la línea de tiempo, reflejando la alta tasa de conexión a las redes Docentes, *CEDIA* y Código Trabajo.

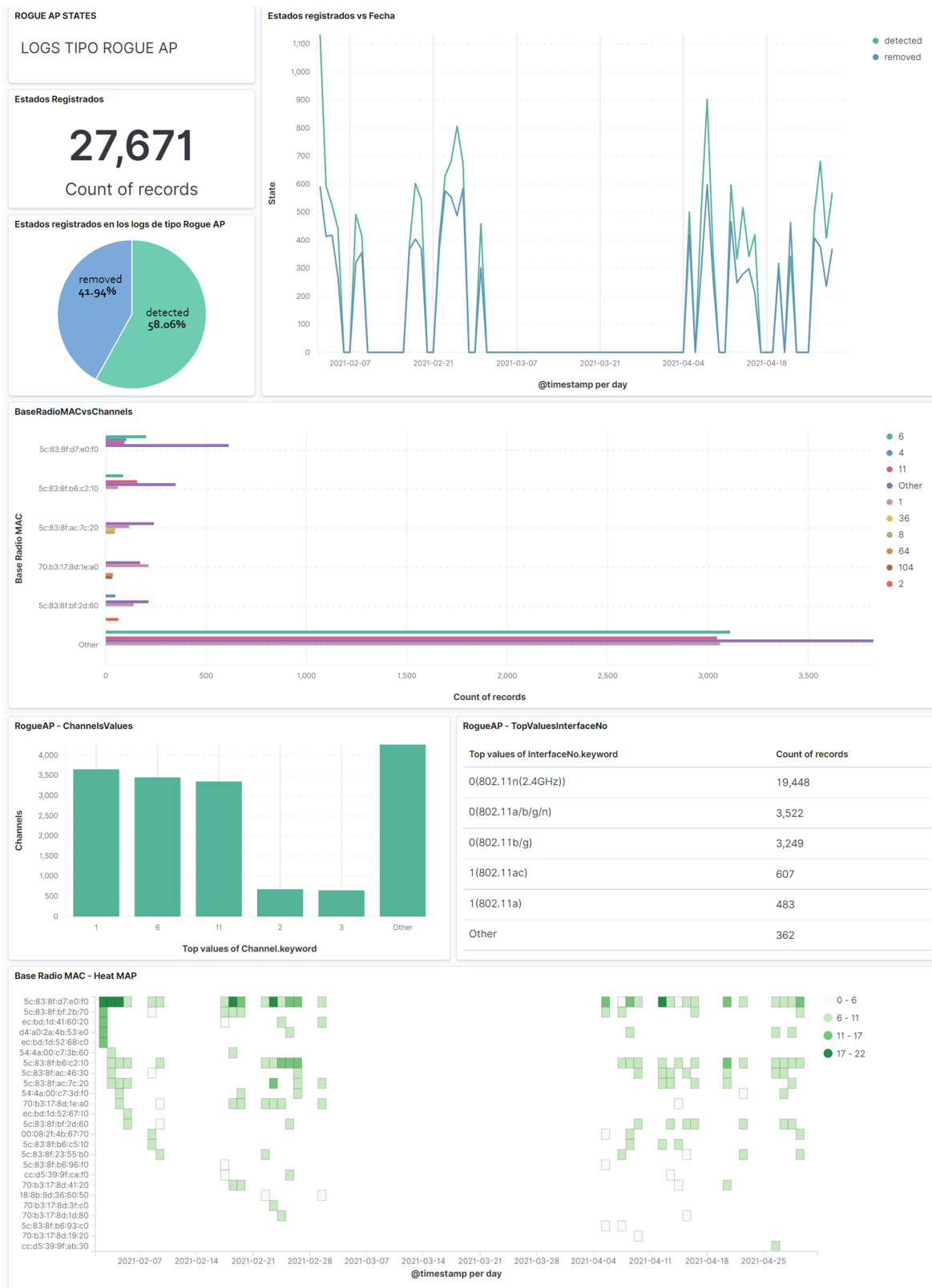


Figura 5.25: *Dashboard* de resultados correspondientes al análisis de los eventos de tipo **Rogue AP**.

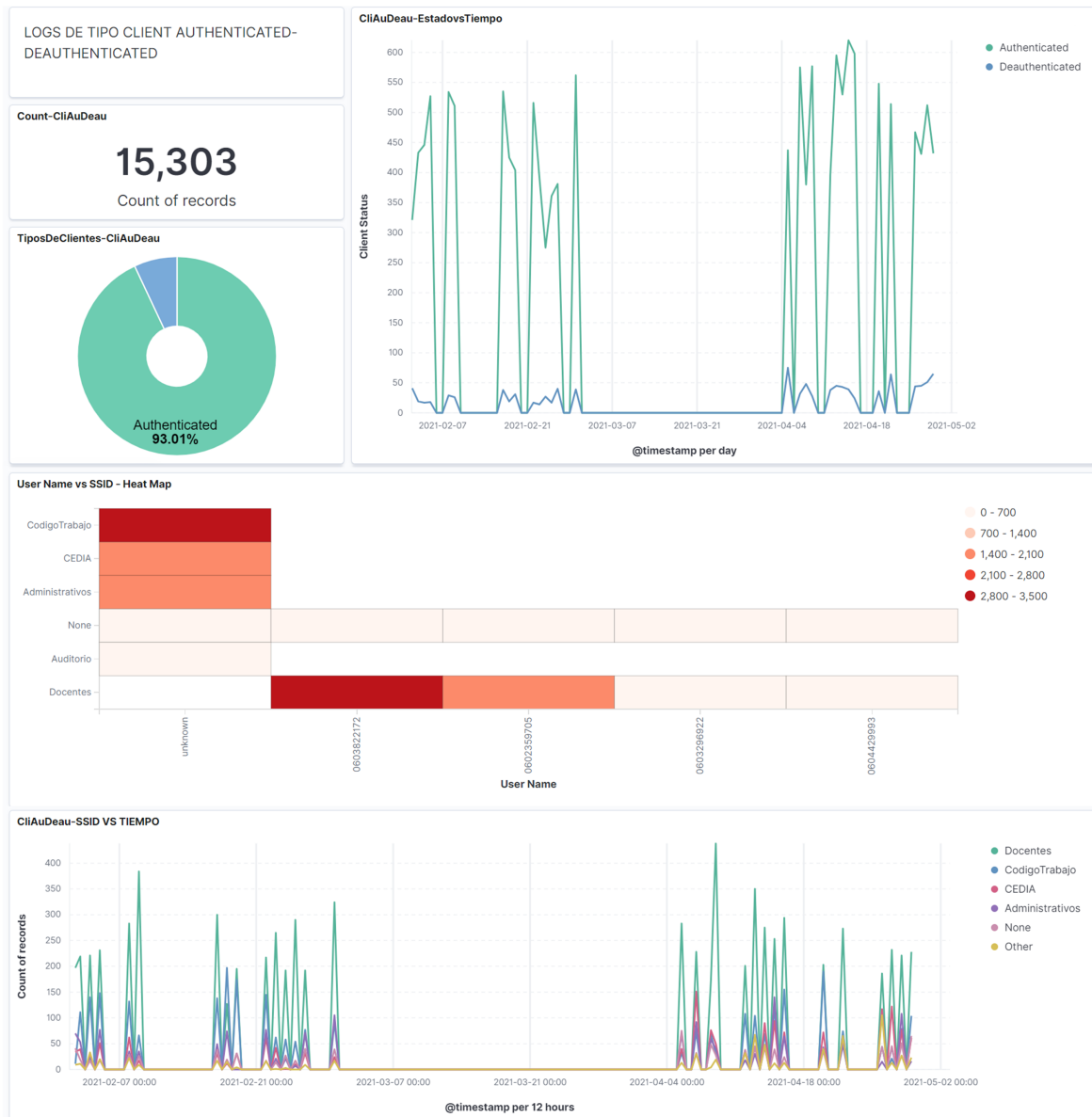


Figura 5.26: *Dashboard* de resultados correspondientes al análisis de los eventos de tipo *Client Authenticated/Deauthenticated*.



CAPÍTULO 6

Conclusiones

En este apartado se realizará las conclusiones del trabajo experimental realizado y una visión de trabajos futuros relacionados a este proyecto.

6.1. Conclusiones

El estado de arte realizado sobre la detección de anomalías, amenazas a la seguridad y las herramientas utilizadas en este ámbito, permitió obtener un conocimiento previo del estado de estos campos. La seguridad dentro de las instituciones o empresas ha sido un punto de relevancia a la hora de crear *frameworks*, debido a la gran cantidad de datos que se manejan y las posibles anomalías o amenazas que se presenten. Los tipos de ataques registrados dentro de una red inalámbrica generalmente son de tipo activo o pasivo; sin embargo, parte de las anomalías detectadas son producto de configuraciones predeterminadas en los equipos o brechas en su arquitectura. Estas brechas o configuraciones permiten a los atacantes visibilizar, manipular o extraer datos. A través del estado del arte realizado, se obtuvo un conocimiento general acerca de las herramientas utilizadas en este ámbito para la detección de anomalías; siendo el algoritmo de clusterización K-Means una opción apropiada para la obtención de resultados congruentes. Esto se debe a sus ventajas con el manejo de grandes conjuntos de datos, su agrupación mediante la identificación de similitudes entre vectores y a los altos porcentajes de detección de anomalías presentados en experimentos pasados.

Todos los eventos producidos en la red quedan registrados en el servidor de *logs*. La información se obtuvo del *WLC* de la *IES* y fue usada como datos de entrada del *framework* desarrollado. De manera puntual, en los *logs* del *WLC* se identificaron varios tipos de eventos: gestión, control, datos, *Rogue AP*, *IDS*, y *Warning*.

El *framework* implementado mediante la utilización de la pila *ELK* y el algoritmo de clusterización K-Means, permitió definir etapas esenciales dentro de la detección de anomalías, como: identificación de eventos, preprocesamiento y análisis de datos. La aplicación de estas etapas, entregaron resultados congruentes que permitieron la posterior aplicación del algoritmo y la entrega de resultados de manera gráfica. Este procedimiento facilitó el análisis de resultados por parte del analista de datos, concluyendo

así que las herramientas utilizadas optimizan el proceso de detección de anomalías en este ámbito. Como se mencionó en los objetivos del presente trabajo de titulación, se realizaron tres experimentos que corroboraron la utilidad de las técnicas de ML como potencial herramienta para la detección de anomalías o amenazas. En el primer experimento, dentro de los eventos de tipo RogueAP, se detectaron dos estados posibles (removed y detected). Esto implica una anomalía, debido a la presencia de eventos que únicamente se detectan y no reciben la aplicación de un procedimiento de análisis o eliminación de manera inmediata. Además, se identificó la utilización de canales de transmisión de datos, tales como el canal 2, que sugieren la presencia de interferencias. Finalmente, según los valores obtenidos en la clusterización, la intensidad de señal y el rendimiento de esta se ve afectada debido a los valores obtenidos para el atributo RSSI, los cuales rondan los -86 dBm. Esto implica una intensidad de señal muy baja, lo cual debe ser tomado en cuenta por el analista de datos.

En el segundo experimento realizado, los resultados obtenidos en la clusterización de los eventos de tipo Client Authenticated/Deauthenticated, arrojaron un alto porcentaje de clientes que logran la autenticación y poseen el UserName unknown. Esto implica una potencial anomalía, ya que este nombre de usuario no cumple con los estándares seguidos por la institución y puede implicar posibles ataques por parte de terceros que no se encuentran autorizados. Además, los resultados de los clústers obtenidos presentaron un alto porcentaje de conexión de este tipo de clientes al SSID CEDIA y CódigoTrabajo. Debido a estos resultados, se realizó un tercer experimento donde la clusterización de las variables UserName y CodigoTrabajo permitió visualizar a profundidad la relación entre usuarios y las diferentes redes, obteniendo un alto porcentaje de conexión por parte de usuarios autorizados al SSID Docentes. Con los resultados obtenidos en estos experimentos, se concluye que el algoritmo de K-Means se encuentra altamente cualificado para la detección de anomalías y amenazas dentro de una red universitaria. Su ejecución permitió verificar el desempeño de la red inalámbrica de la UNACH, identificando accesos de usuarios que incumplen la normativa institucional, detección de AP no autorizados o anomalías en el desempeño de la red, tales como interferencias.

Todas las anomalías detectadas usando el *framework* implementado, fueron presentadas a través de *dashboards* dentro de la herramienta Kibana. Los gráficos relacionales entre campos o en la línea de tiempo mostraron que las anomalías detectadas por la clusterización son correctas; lo que indica la fiabilidad de los resultados entregados por el *framework*. Además, la finalidad de estos *dashboards*, es que el analista de datos conozca el estado de la red de manera gráfica, visualizando posibles anomalías al realizar un análisis general de los campos.

Los resultados obtenidos en este trabajo experimental indican que las técnicas de ML son una herramienta potente para la aplicación en el ámbito de la seguridad. Cabe recalcar que la implementación del *framework* descrito en este documento se vio limitado por la falta de datos dentro de la línea temporal; sin embargo, los resultados obtenidos pueden ser utilizados como soporte para una futura detección de anomalías por parte del analista de datos.

6.2. Trabajos Futuros

Se plantea la completa automatización del *framework*, desarrollando un mecanismo que permita la detección automática de una serie de eventos que se consideren potenciales amenazas a la seguridad, tales como eventos de tipo Rogue AP o IDS. Esto evitaría la intervención del analista de datos en las etapas intermedias. Además, la aplicación en tiempo real del *framework* permitiría la aplicación de



algoritmos que permitan anticipar eventos que comprometan el desempeño de la red. Esto evitaría, por ejemplo, la aglomeración de usuarios en una red específica o la colocación de un **AP** extra que evite la saturación del **AP** principal, disminución de calidad de la señal de transmisión o variación en las tasas de subida y bajada de datos.



Instalación y configuración del *software*.

En este anexo se detalla la instalación y configuración de las herramientas de *software* utilizadas en el desarrollo de este proyecto.

A.1. Instalación y configuración de la pila ELK

Para la instalación de esta pila, se tomó como referencia la información presentada en la página oficial del proyecto [47], y como complemento las guías de instalación facilitadas en [48] y [49].

A.1.1. Prerrequisitos

El principal prerrequisito para la instalación de esta pila es la instalación de los paquetes de desarrollo de Java 8 y 11 en cada uno de los servidores, mediante el comando de instalación presentado en el Listado A.1.

```
# apt install openjdk-8-jdk
# apt install openjdk-11-jdk
```

Listado A.1: Instalación de paquetes de Java

A.1.2. Instalación y configuración de Elasticsearch

Mediante los comandos presentados en el Listado A.2 se instala todas las dependencias necesarias para la herramienta, además de la importación de la clave de GPG pública de Elasticsearch en APT y lista de fuentes de Elastic al directorio `sources.list.d`. A continuación se realiza una actualización que permita la instalación de Elasticsearch.

```
# apt update
# apt install apt-transport-https ca-certificates wget
#wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
# sh -c 'echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" > /etc/
    apt/sources.list.d/elastic-7.x.list'
# apt update
```



```
# apt install elasticsearch
```

Listado A.2: Instalación de Elasticsearch.

A.1.2.1. Edición del archivo de configuración principal de Elasticsearch

Con el comando presentado en A.3, se realiza la configuración del archivo `.yml` de Elasticsearch, donde se especifica los siguientes parámetros: `network.host: 127.0.0.1`, `http.host: 10.2.3.174` y `http.port:9200`. Esto permite el ingreso al servicio de manera externa, además de la asignación de un puerto específico para la comunicación con el resto de servidores.

```
#nano /etc/elasticsearch/elasticsearch.yml
```

Listado A.3: Configuración del archivo principal de Elasticsearch.

A.1.2.2. Iniciación del servicio Elasticsearch

Finalizada la instalación y configuración, se habilita el servicio para que inicie automáticamente, además se verifica el estado actual de la herramienta. Para este proceso se aplica los comandos definidos en A.4.

```
#systemctl daemon-reload
#systemctl enable elasticsearch.service
#systemctl start elasticsearch.service
#systemctl status elasticsearch.service
#curl http://[MASTER_IP]:9200
```

Listado A.4: Activación e inicio de Elasticsearch.

A.1.3. Instalación y configuración de Logstash

Debido a que las herramientas se encuentran distribuidas en diferentes servidores, es necesaria la importación de la clave [GNU Privacy Guard \(GPG\)](#) de Elasticsearch y la lista de fuentes de Elastic al directorio `sources.list.d`. De esta manera, se procede a la instalación de la herramienta mediante los comandos presentados en en Listado A.5.

```
#apt update
#apt install apt-transport-https ca-certificates wget
#wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
#sh -c 'echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" > /etc/
apt/sources.list.d/elastic-7.x.list'
#apt update
#apt install logstash
```

Listado A.5: Instalación de Logstash.

A.1.3.1. Edición del archivo de configuración principal de Logstash

Mediante el Listado A.6 se realiza la dirección del `host` que posee Logstash y los puertos a utilizar.

```
#nano /etc/logstash/logstash.yml
```

Listado A.6: Configuración del archivo principal de Logstash.



A.1.3.2. Iniciación del servicio Logstash

Los comandos presentados en el Listado A.7 permiten la iniciación del servicio.

```
#systemctl daemon-reload
#systemctl enable logstash.service
#systemctl start logstash.service
#systemctl status logstash.service
```

Listado A.7: Activación e inicio de Logstash.

A.1.4. Instalación y configuración de Kibana

Como se mencionó anteriormente en las Secciones y , en esta etapa se realizó la importación de la clave [GPG](#) de Elasticsearch y la lista de fuentes de Elastic al directorio `sources.list.d`, además de realizar el proceso de instalación mediante el Listado presentado en A.8.

```
#apt update
#apt install apt-transport-https ca-certificates wget
#wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
#sh -c 'echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" > /etc/
apt/sources.list.d/elastic-7.x.list'
#apt update
#apt install kibana
```

Listado A.8: Instalación de Kibana.

A.1.4.1. Edición del archivo de configuración principal de Kibana

A través del Listado mostrado en A.9, se realiza la declaración de una serie de configuraciones como: el puerto a utilizar por el servidor que aloja Kibana (`server.port: 5601`), la dirección [IP](#) asignada al servidor (`server.host: 10.2.1.90`) y finalmente la dirección [IP](#) correspondiente a Elasticsearch junto a su puerto (`elasticsearch.hosts:["http://10.2.3.174:9200"]`).

```
#nano /etc/kibana/kibana.yml
```

Listado A.9: Configuración del archivo principal de Kibana.

A.1.4.2. Iniciación del servicio Kibana

De manera semejante a las dos herramientas descritas con anterioridad, Kibana se define como un servicio, que puede ser habilitado, iniciado o detenido. En el Listado A.10 se indica los comandos correspondientes al proceso de habilitación, iniciación y verificación del estado de Kibana.

```
#systemctl daemon-reload
#systemctl enable kibana.service
#systemctl start kibana.service
#systemctl status kibana.service
```

Listado A.10: Activación e inicio de Kibana.



A.1.5. Instalación y configuración de NGINX

Para el acceso de manera externa a la pila ELK, se utilizó el servidor web *open source* NGINX, el cual es instalado en el servidor correspondiente a Kibana. En el Listado A.11 se muestra los comandos necesarios para su instalación y verificación de puertos de abiertos a utilizar para la comunicación con el resto de servidores pertenecientes a la pila.

```
#apt update
#apt install nginx
#ufw app list
#ufw status
#systemctl status nginx
```

Listado A.11: Instalación y configuración de NGINX.

En el Listado A.12, se indica el proceso de configuración de usuario y contraseña, con la finalidad de proteger el ingreso a la herramienta por parte de terceros. Mediante *openssl* se crea un usuario administrativo de Kibana con su contraseña correspondiente. Esta información será almacenada en el archivo `htpasswd.kibana`, y establecerá que Nginx requiera esta información al establecer la conexión.

```
#echo "admin:'openssl passwd -apr1 admin'" | sudo tee -a /etc/nginx/htpasswd.kibana
```

Listado A.12: Proceso de declaración de usuario y contraseña.

A continuación, mediante el comando presentando en el Listado A.13, se crea un archivo de configuración llamado `kibana`, con la finalidad de realizar la declaración de las direcciones IP y puertos a utilizar.

```
#nano /etc/nginx/sites-available/kibana
```

Listado A.13: Proceso de declaración de usuario y clave para el ingreso.

El archivo de configuración mencionado en la Sección A.13, posee el bloque de código presentado en A.14. En este, Nginx dirige tráfico *Hypertext Transfer Protocol* (HTTP) de su servidor a la aplicación de Kibana. Este proceso lo realiza a través de la dirección y puerto `10.2.1.90:5601`. También se configura Nginx para leer el archivo `kibana` y requerir la autenticación básica.

```
server {
    listen 80;

    server_name 10.2.1.90;

    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.kibana;

    location / {
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

```
}  
}
```

Listado A.14: Configuración del archivo de enrutamiento de tráfico a Kibana.

Finalmente se habilitó la configuración creando un enlace simbólico al directorio `sites-enabled`, permitiendo el acceso al servicio (ver Figura A.1)

```
#rm /etc/nginx/sites-enabled/default  
#ln -s /etc/nginx/sites-available/kibana /etc/nginx/sites-enabled/kibana  
#systemctl restart nginx
```

Listado A.15: Habilitación y reinicio del servicio.

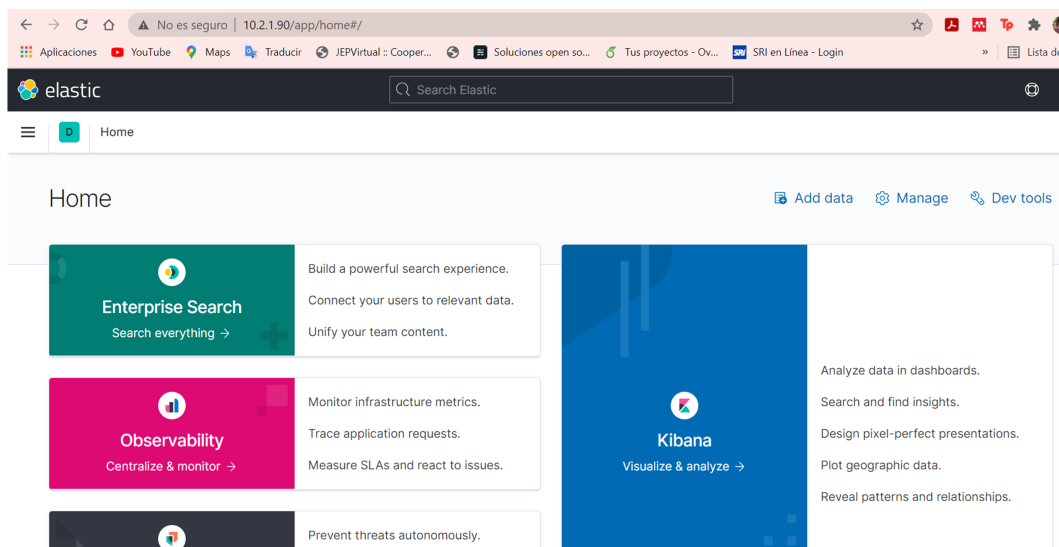


Figura A.1: Ingreso a Kibana mediante el proxy inverso Nginx



Aprendizaje automático con Weka y creación de *dashboards*

B.1. Aprendizaje automático con Weka

Inicialmente se procede a subir el archivo correspondiente a los datos normalizados, invocando el cuadro de diálogo que permita la declaración del formato de fecha y separador utilizado. Una vez subido el *dataset*, en la interfaz gráfica se visualiza las variables que conforman el tipo de evento, además de información de interés como el valor máximo, mínimo, media, o gráficas de barras que representan el estado de cada una de ellas, como se presenta en la Figura B.1.

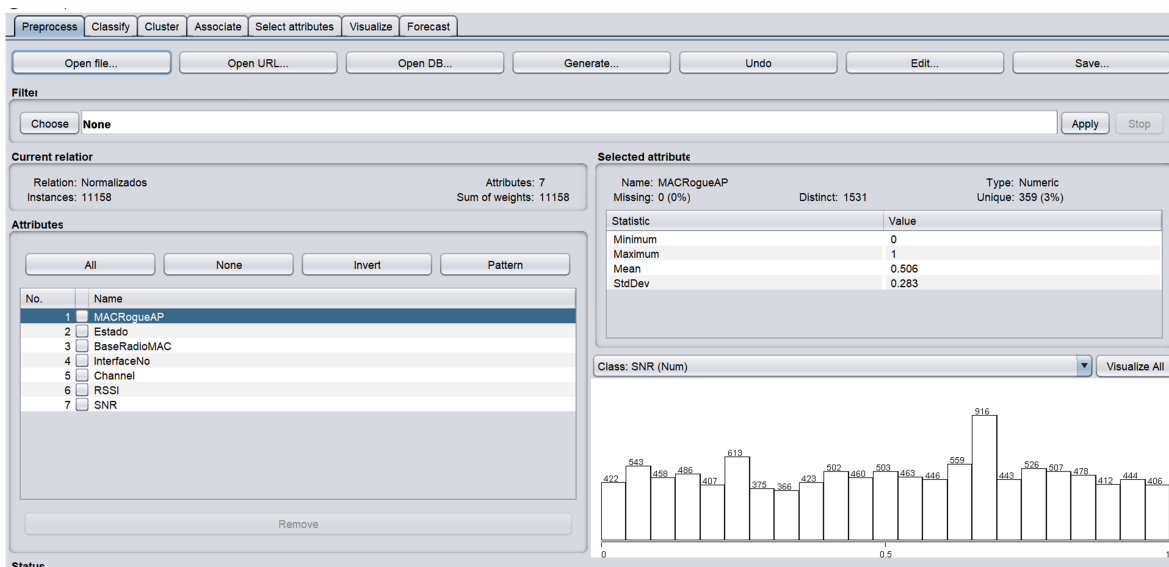


Figura B.1: Variables en la interfaz gráfica de Weka

A continuación, en la parte superior se sitúa la pestaña correspondiente a los algoritmos de clusterización, donde en la barra de búsqueda se selecciona el algoritmo de K-Means. En sus propiedades,

es necesaria la declaración del número de clústers obtenidos mediante el método de Elbow, por lo que una vez finalizado este procedimiento, se realiza la aplicación del algoritmo. Como resultado, este devuelve mediante una interfaz gráfica los parámetros correspondientes a cada uno de los clústers y finalmente el porcentaje de datos que corresponde a cada uno, como se aprecia en la Figura B.2.

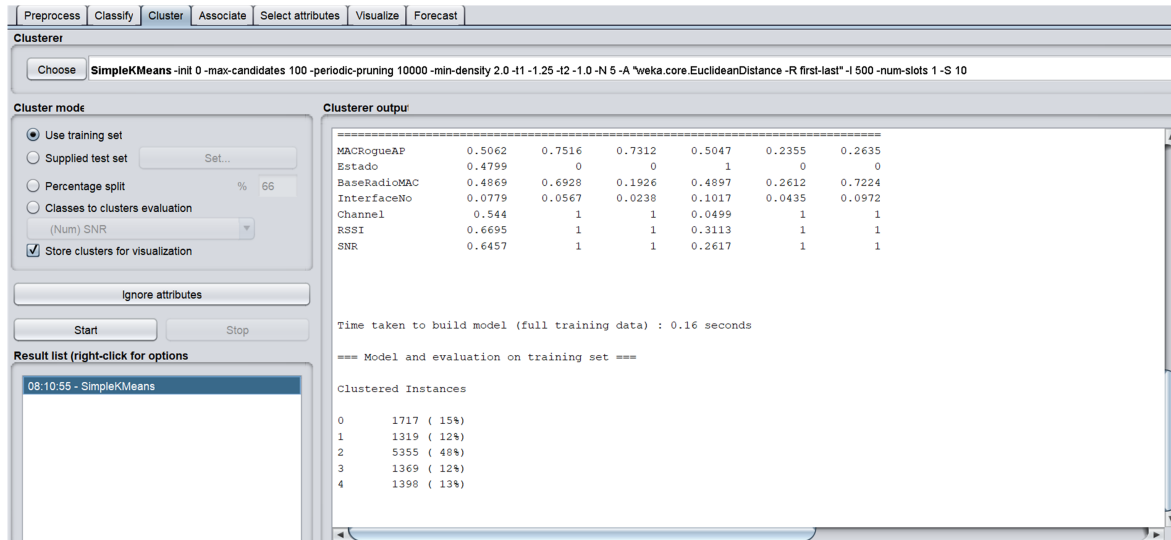


Figura B.2: K-Means en Weka

B.2. Creación de *dashboards*

Para la visualización de resultados, se aplicó la herramienta *Lens* disponible en Kibana. Esta es una interfaz de usuario intuitiva que permite la generación de diferentes tipos de gráficos mediante la selección de la variable o variables a relacionar, como se presenta en la Figura B.3.

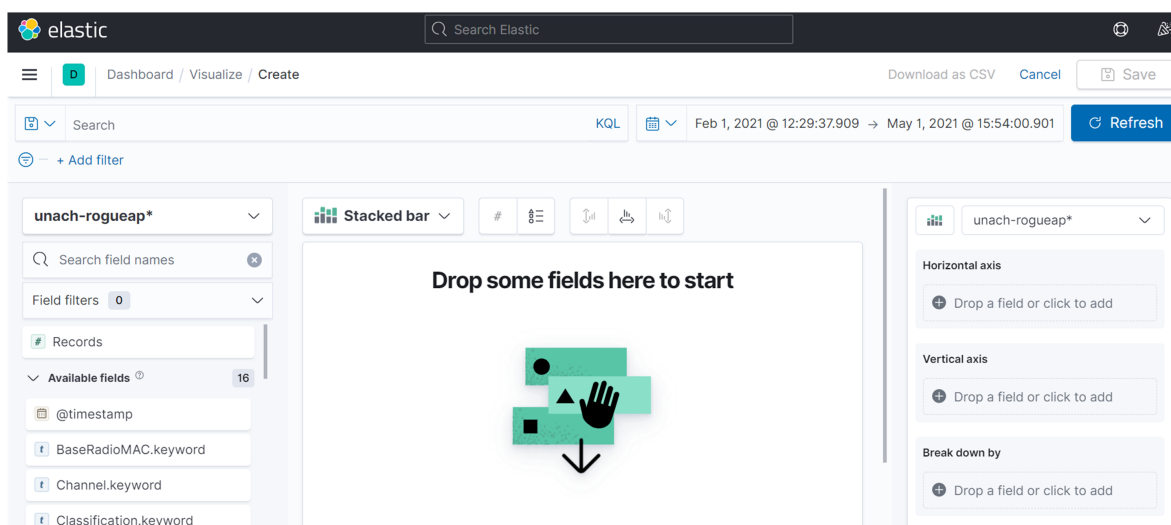


Figura B.3: Interfaz gráfica de la herramienta Lens



Dashboards de eventos identificados



Figura C.1: Variables manejadas por los eventos de tipo Client Authenticated/Deauthenticated.

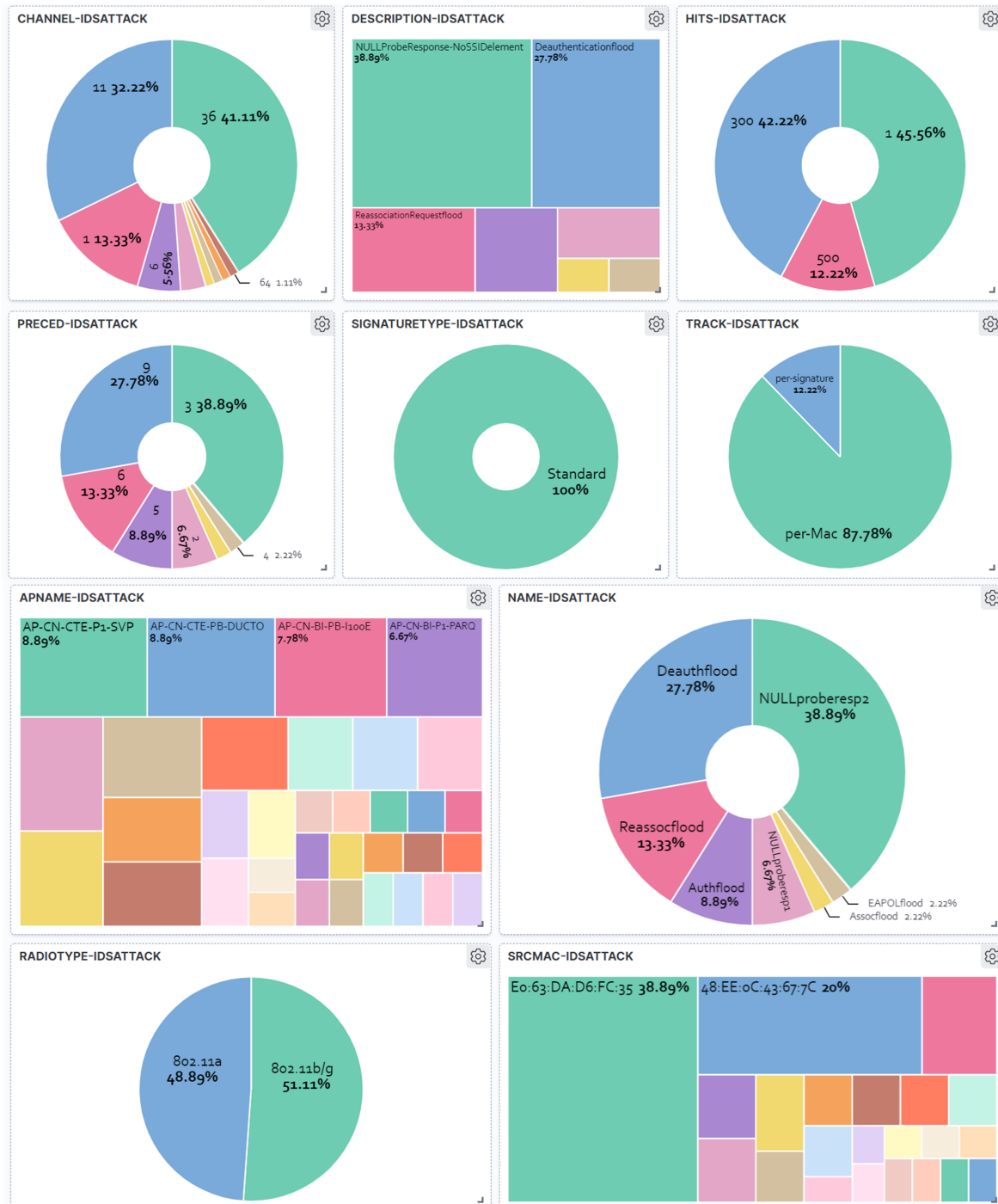


Figura C.2: Variables manejadas por los *logs* de tipo IDSAttack.

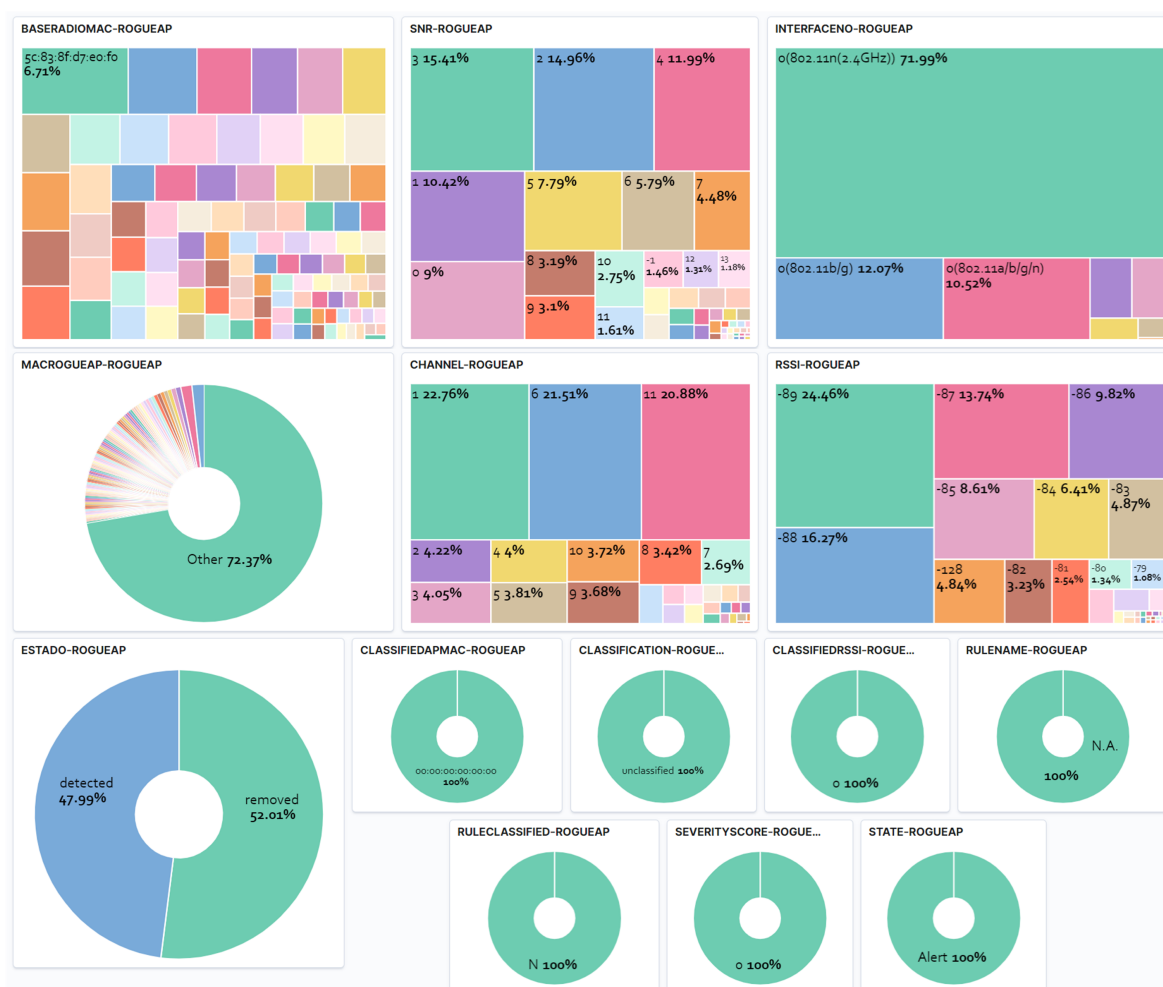


Figura C.3: Variables manejadas por los eventos de tipo RogueAP

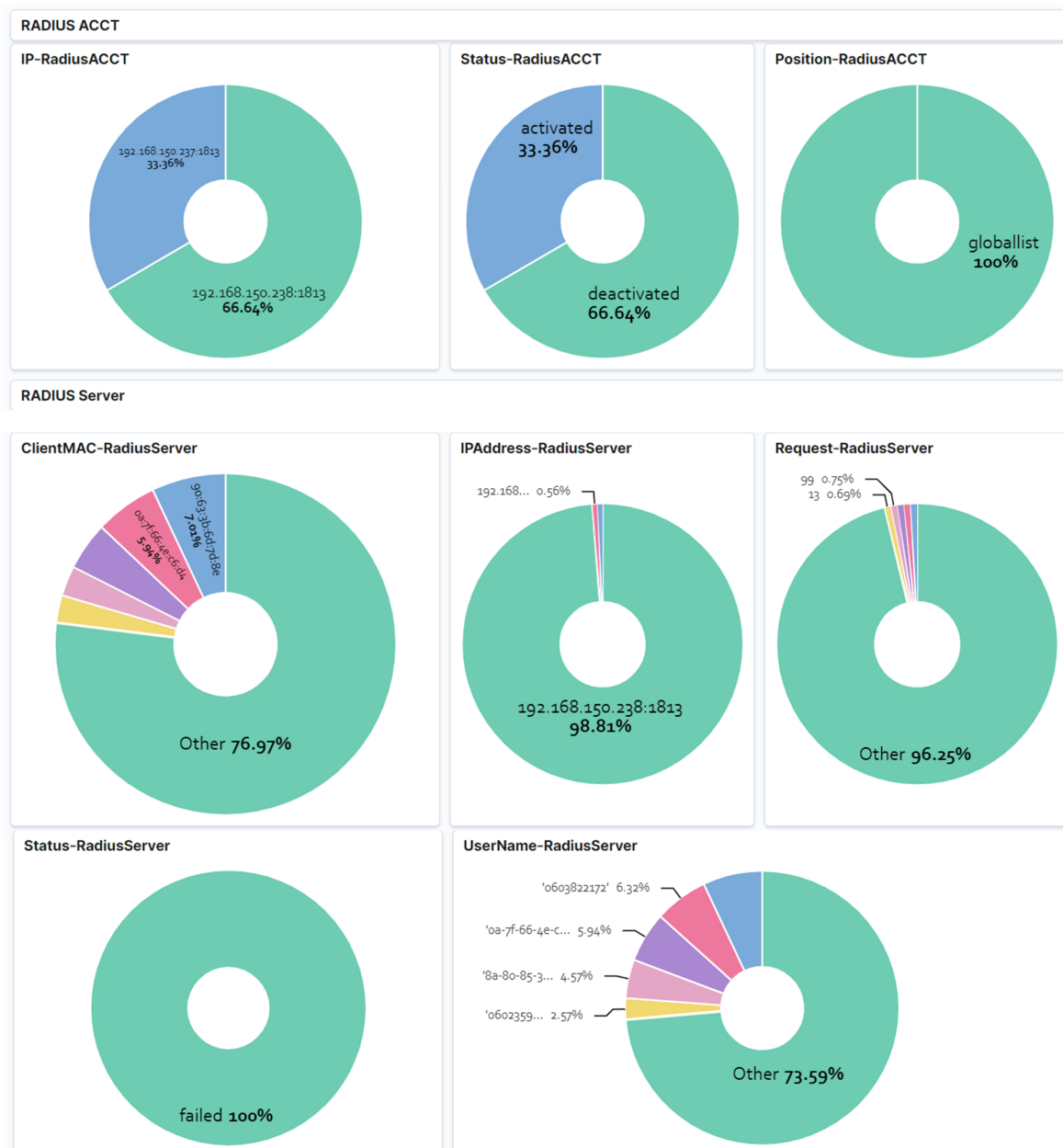


Figura C.4: Variables manejadas por los *logs* de tipo **RADIUS**



Bibliografía

- [1] A. d. C. A. Estupiñan, J. A. Pulido, y J. A. B. Jaime, “Análisis de riesgos en seguridad de la información,” *Ciencia, innovación y tecnología*, vol. 1, pp. 40–53, 2013.
- [2] J. M. Pena, J. A. Lozano, y P. Larranaga, “An empirical comparison of four initialization methods for the k-means algorithm,” *Pattern recognition letters*, vol. 20, num. 10, pp. 1027–1040, 1999.
- [3] R. Ríos y J. R. Fermin, “Análisis de tráfico de una red local universitaria,” *Télématique*, vol. 8, num. 2, pp. 15–27, 2009.
- [4] T. Thomas, A. P. Vijayaraghavan, y S. Emmanuel, *Machine learning approaches in cyber security analytics*. Springer, 2020.
- [5] N. V. Avellán Zambrano y M. F. Zambrano Bravo, “Ciberseguridad y su aplicación en las instituciones de educación superior públicas de manabí,” Master’s thesis, Calceta: ESPAM MFL, 2019.
- [6] “X.1205 : Aspectos generales de la ciberseguridad.” [En línea]. Disponible: <https://www.itu.int/rec/T-REC-X.1205-200804-I/es>
- [7] S. E. Salama, M. I. Marie, L. M. El Fangary, y Y. K. Helmy, “Web server logs preprocessing for web intrusion detection,” *Comput. Inf. Sci.*, vol. 4, num. 4, pp. 123–133, 2011.
- [8] D. Ignacio Conde, “Plataforma de evaluación de rendimiento de tráfico de red basada en la pila elk,” 2020. [En línea]. Disponible: <https://ruc.udc.es/dspace/handle/2183/25608>
- [9] V. H. Gálvez Caza, “Propuesta de mejora en la seguridad de redes inalámbricas, utilizando los estándares definidos por nist. caso de estudio: red wifi de la ies.” Master’s thesis, PUCE, 2017.
- [10] S. D. Barrera Morales, “Sistema de análisis y monitoreo para la red inalámbrica de la empresa distry-tex,” B.S. thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, 2009.
- [11] W. Rodríguez Aburto y P. J. Castellón Mena, “Propuesta de detección y mitigación de ataques de denegación de servicios en las redes institucionales dgi,” Ph.D. dissertation, Universidad Nacional de Ingeniería, 2019.
- [12] M. Soriano, “Seguridad en redes y seguridad de la información,” *Obtenido de http://improvet.cvut.cz/project/download/C2ES/Seguridad_de_Red_e_Informacion.pdf*, 2014.
- [13] T. Dunning y E. Friedman, *Practical machine learning: a new look at anomaly detection*. .O’Reilly Media, Inc.", 2014.



-
- [14] “aprendizaje | Definición | Diccionario de la lengua española | RAE - ASALE.” [En línea]. Disponible: <https://dle.rae.es/aprendizaje>
- [15] “Client Association Packet Flow (Cisco Wireless LAN Controllers).” [En línea]. Disponible: <http://what-when-how.com/deploying-and-troubleshooting-cisco-wireless-lan-controllers/client-association-packet-flow-cisco-wireless-lan-controllers/>
- [16] A. Singh, N. Thakur, y A. Sharma, “A review of supervised machine learning algorithms,” in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. Ieee, 2016, pp. 1310–1315.
- [17] M. Syakur, B. Khotimah, E. Rochman, y B. D. Satoto, “Integration k-means clustering method and elbow method for identification of the best customer profile cluster,” in *IOP Conference Series: Materials Science and Engineering*, vol. 336, num. 1. IOP Publishing, 2018, p. 012017.
- [18] P. Barford, J. Kline, D. Plonka, y A. Ron, “A signal analysis of network traffic anomalies,” in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, 2002, pp. 71–82.
- [19] A. Kind, M. P. Stoecklin, y X. Dimitropoulos, “Histogram-based traffic anomaly detection,” *IEEE Transactions on Network and Service Management*, vol. 6, num. 2, pp. 110–121, 2009.
- [20] K. Limthong y T. Tawsook, “Network traffic anomaly detection using machine learning approaches,” in *2012 IEEE Network Operations and Management Symposium*. IEEE, 2012, pp. 542–545.
- [21] “Free and Open Search: The Creators of Elasticsearch, ELK & Kibana.” [En línea]. Disponible: <https://www.elastic.co/es/what-is/elk-stack>.
- [22] “Logstash: Recopila, parsea y transforma logs.” [En línea]. Disponible: <https://www.elastic.co/es/logstash/>
- [23] “Elasticsearch: El motor de búsqueda y analítica distribuido oficial.” [En línea]. Disponible: <https://www.elastic.co/es/elasticsearch/>
- [24] “¿Qué es Elasticsearch?” [En línea]. Disponible: <https://www.elastic.co/es/what-is/elasticsearch>
- [25] “¿Qué es Kibana?” [En línea]. Disponible: <https://www.elastic.co/es/what-is/kibana>
- [26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, y I. H. Witten, “The weka data mining software: an update,” *ACM SIGKDD explorations newsletter*, vol. 11, num. 1, pp. 10–18, 2009.
- [27] Z. Markov y I. Russell, “An introduction to the weka data mining system,” *ACM SIGCSE Bulletin*, vol. 38, num. 3, pp. 367–368, 2006.
- [28] “scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation.” [En línea]. Disponible: <https://scikit-learn.org/stable/>
- [29] G. Fernandes, J. J. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, y M. L. Proença, “A comprehensive survey on network anomaly detection,” *Telecommunication Systems*, vol. 70, num. 3, pp. 447–489, 2019.
-



-
- [30] T. Violeta Vallejo de León Asesorado por el Ing MsEE Enrique Edmundo Ruiz Carballo, “Vulnerabilidades y niveles de seguridad de redes Wi-Fi,” 2019.
- [31] J. C. Cruz, A. Z. López, C. A. Cruz, y J. V. Cortez, “Análisis de ataques de red del tipo dhcp spoofing, tcp syn flood y paquetes malformados,” *Pistas Educativas*, vol. 39, num. 128, 2018.
- [32] B. Carrión Ramírez, “Diseño e implementación de una solución de gestión centralizada de logs de aplicaciones, sistemas y dispositivos basada en logstash,” 2019.
- [33] N. Elmrabit, F. Zhou, F. Li, y H. Zhou, “Evaluation of Machine Learning Algorithms for Anomaly Detection,” *International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2020*, jun 2020.
- [34] A. Valencia Peral, “Técnicas de aprendizaje automático para la detección de ataques en el tráfico de red,” 2019.
- [35] M. Nawir, A. Amir, N. Yaakob, y O. B. Lynn, “Effective and efficient network anomaly detection system using machine learning algorithm,” *Bulletin of Electrical Engineering and Informatics*, vol. 8, num. 1, pp. 46–51, 2019.
- [36] M. L. Ambuludi Torres, “Implementación de técnicas de machine learning para la detección de anomalías basados en nids.” B.S. thesis, 2019.
- [37] Á. Martín Fidalgo, “Detección de anomalías en red utilizando técnicas de Machine Learning,” jul 2017. [En línea]. Disponible: <https://e-archivo.uc3m.es/handle/10016/27749>
- [38] “Ubuntu Server 20.04 LTS: stability, security and more | Ubuntu.” [En línea]. Disponible: <https://ubuntu.com/blog/ubuntu-server-20-04>
- [39] “Date field type | Elasticsearch Guide [7.14] | Elastic.” [En línea]. Disponible: <https://www.elastic.co/guide/en/elasticsearch/reference/current/date.html>
- [40] A. A. ARCIA PLUA, “Análisis de tráfico de datos en la capa de enlace de redes lan, para la detección de posibles ataque o intrusiones sobre tecnologías ethernet y wifi 802.11 en la carrera de ingeniería en sistemas computacionales de la universidad estatal del sur de manabí,” B.S. thesis, Jipijapa. UNESUM, 2021.
- [41] M. Barrionuevo, M. Lopresti, N. C. Miranda, y M. F. Piccoli, “Un enfoque para la detección de anomalías en el tráfico de red usando imágenes y técnicas de computación de alto desempeño,” in *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*, 2016.
- [42] “¿Qué es un IDS o Intrusion Detection System? | Clavei.” [En línea]. Disponible: <https://www.clavei.es/blog/que-es-un-ids-o-intrusion-detection-system/>
- [43] J. Pérez Sifre, “Ids de red para la detección de ataques sobre ssh y ftp,” 2020. [En línea]. Disponible: <http://rua.ua.es/dspace/handle/10045/107579>
- [44] “Qué es un servidor Radius y cómo funciona.” [En línea]. Disponible: <https://www.redeszone.net/2017/06/02/servidor-radius-funciona/>
-



- [45] D. Gómez Barquero, “Análisis y optimización de redes wi-fi,” 2021. [En línea]. Disponible: <https://riunet.upv.es/handle/10251/167530>
- [46] M. N. Borenovic y A. M. Neskovic, “Comparative analysis of rssi, snr and noise level parameters applicability for wlan positioning purposes,” in *IEEE EUROCON 2009*. IEEE, 2009, pp. 1895–1900.
- [47] “Installing the Elastic Stack | Installation and Upgrade Guide [7.14] | Elastic.” [En línea]. Disponible: <https://www.elastic.co/guide/en/elastic-stack/current/installing-elastic-stack.html>
- [48] “Tutorial para instalar y configurar ElasticStack: Elasticsearch, Logstash, Kibana, Beats | Guillermo Alvarado.” [En línea]. Disponible: <https://galvarado.com.mx/post/cómo-instalar-elasticstack-elasticsearch-logstash-kibana-beats/>
- [49] “Installing the ELK Stack on Windows | Logz.io.” [En línea]. Disponible: <https://logz.io/blog/elk-stack-windows/>